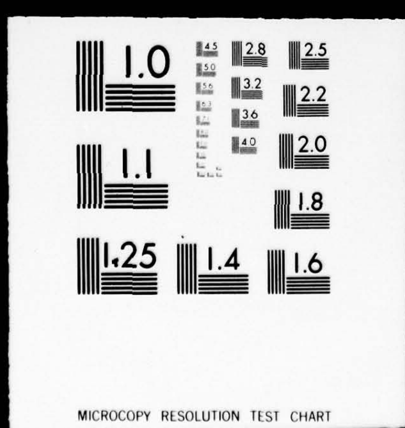
AD-A074 972 VIRGINIA POLYTECHNIC INST AND STATE UNIV BLACKSBURG --ETC F/G 20/4
USER'S GUIDE FOR A SYSTEM OF COMPUTER PROGRAMS TO PREDICT FLOWS--ETC(U)
NOV 78 M C HYMAN, S C HOULIHAN, J A HILL
UNCLASSIFIED VPI-AERO-096 NL

1 OF 4
ADA 074972

1 OF 4

ADA 074972



VPI-AERO-096

User's Guide For A System Of Computer Programs To Predict Flows Over And In The Near Wake Behind Axisymmetric, Self-Propelled Bodies

by

M. C. Hyman, S. C. Houlihan, J. A. Hill,
D. L. Dwoyer and C. H. Lewis

Department of Aerospace and Ocean Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

November 1978

1

6

USER'S GUIDE FOR A SYSTEM OF COMPUTER PROGRAMS TO PREDICT FLOWS OVER AND IN THE NEAR WAKE BEHIND AXISYMMETRIC, SELF-PROPELLED BODIES,

by

10 M. C. Hyman, S. C. Houlihan, J. A. Hill, D. L. Dwoyer ~~and~~ C. H. Lewis

12 296

14 VPI-AERO-Ø96

Department of Aerospace and Ocean Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

11 November 1978

406 922

JOB

ABSTRACT

A system of computer programs has been developed to predict flows around and in the near wake behind an axisymmetric body including a propeller at the tail of the vehicle. The system is capable of calculating laminar and/or turbulent, attached and separated flows. Interactive boundary-layer theory is used to predict the flow on the nonseparating region over the body and the full Navier-Stokes equations are solved for the separated, propeller and near wake flow regions.

Details of the theory and results are included in a separate engineering report. The programs, subroutines, variables in common and input and output data are described. Listings of the program code, input and output data for a sample case are included.

FORWARD

The following system of computer programs has been developed for use on the IBM 370, model 158 computer, for CMS under VM. For this reason the exec procedures required to execute this system have also been included to aid in the conversion of the system in its present form to a form compatible with that in use at a receiving facility.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

INTRODUCTION

The purpose of this system of programs is the prediction of viscous flow over and in the near wake behind axisymmetric bodies, particularly submarine-like bodies. The effects of an operating propeller and separated flow have been included in the procedure. The system procedure is basically the execution of four segments. The first is the development of a transformed or nonseparating body which allows a boundary-layer solution to be calculated. The second segment is the solution of the interactive boundary-layer over the body. The third segment is the development of the perturbed flowfield, used in the fourth and final segment, the solution of the afterbody and near wake with the propeller.

DESCRIPTION OF SYSTEM

The system is composed of a number of programs which serve not only to obtain a solution, but to as great a degree as possible automate the calculation of the solution. The first segment of the system is composed of support programs included specifically to perform this task. The general body coordinates are input and these programs calculate a transformed set of coordinates that allow a nonseparating boundary-layer solution to be obtained. The basis of the transformation is in the nature of the flow over the afterbody. The original coordinates are tested to determine if the flow separates over the afterbody. The test is based on the inviscid pressure distribution over the body and was empirically developed from repeated trials on a number of test configurations. From these trials it was determined that the boundary layer on the afterbody would not remain attached if the inviscid pressure coefficient exceeded a value of 0.2. Therefore, an inviscid calculation is performed to give the pressures over the afterbody and, if the criterion for separation is met, the body is modified until a nonseparating configuration is found.

The next segment of the system is an interactive boundary-layer solution about the transformed body. The interactive solution employs the VPI&SU Incompressible Boundary-Layer Program and the Hess Potential Flow Program to obtain a viscous/inviscid iterated solution. The iteration procedure is as follows: A boundary-layer solution is first obtained by matching the boundary-layer edge pressure with a "first guess" inviscid pressure distribution. Coordinates of the displacement body

that result from the boundary-layer solution are placed in a separate coordinate data set. The inviscid pressure distribution is then found about the displaced body by the potential flow program and becomes the next guess for the pressure distribution. This guess is then used by the boundary-layer program to yield another, corresponding displaced body and the sequence is continued. The iterative process that is developed is continued for several iterations, the number of which is dependent on the shape of the afterbody and the accuracy of the initial guess. With the current system, the first guess is the inviscid pressure distribution about the original body obtained from the potential flow code. It has been found that the use of this initial guess requires that only one iteration was needed when solving for the flows about bodies with straight tails and from 4-6 iterations were needed for straight-tapered or curved tails.

The elliptic nature of the Navier-Stokes equations requires that full (surrounding), boundary conditions be supplied, as well as a first guess of the flowfield in the region treated. The downstream boundary lies in the wake of the body and the corresponding boundary conditions are that $\partial^2\xi/\partial x^2$ and $\partial^2\psi/\partial x^2$ go to zero. Since the region that is treated in the Navier-Stokes program, in this case a "box" of grid points, encompasses a much greater area than that of the viscous region alone, the full viscous-inviscid velocity profile is required at the upstream edge. The interactive boundary-layer solution provides conditions for the viscous region of the upstream boundary, and conditions in the inviscid region are obtained by employing the potential flow program to calculate the outer flowfield about the final displaced body. At the

joining of the two velocity profiles, the viscous and inviscid regions, a composite expansion is used to ensure a continuous transition. The initial flowfield is thus an inviscid velocity field and is used inside the Navier-Stokes program to generate the initial stream function throughout the Navier-Stokes region. The upper boundary conditions are simply the inviscid velocities obtained from the perturbed flowfield.

The final segment of the system is the solution of the propeller, separated and near wake regions of the flow which is accomplished through the use of the APLNS program. The Navier-Stokes program iterates on the stream function, which, on the first iteration, is the stream function about the inviscid, perturbed body described above. An Alternating Difference Implicit method (ADI) is used for calculating new stream functions in the Navier-Stokes (N-S) region. After NTS time steps, the solution is assumed to reach convergence and the results are printed. For many high speed problems the time step (Δt) must be particularly small, and hence the number of time steps required for convergence may be considerable.

There is no final convergence criterion other than the specified total number of time steps (NTS).

The logical flow of the system may be followed in Fig. 1. Figure 2 shows the corresponding sequence of program usage in the system. This user's guide contains a brief overview of the system augmented by a more complete and detailed description of the system and its component programs. The system is tied together by WAKE EXEC, an exec procedure which executes the segment procedures. See Appendix A, Table 1.1 for a listing of WAKE EXEC.

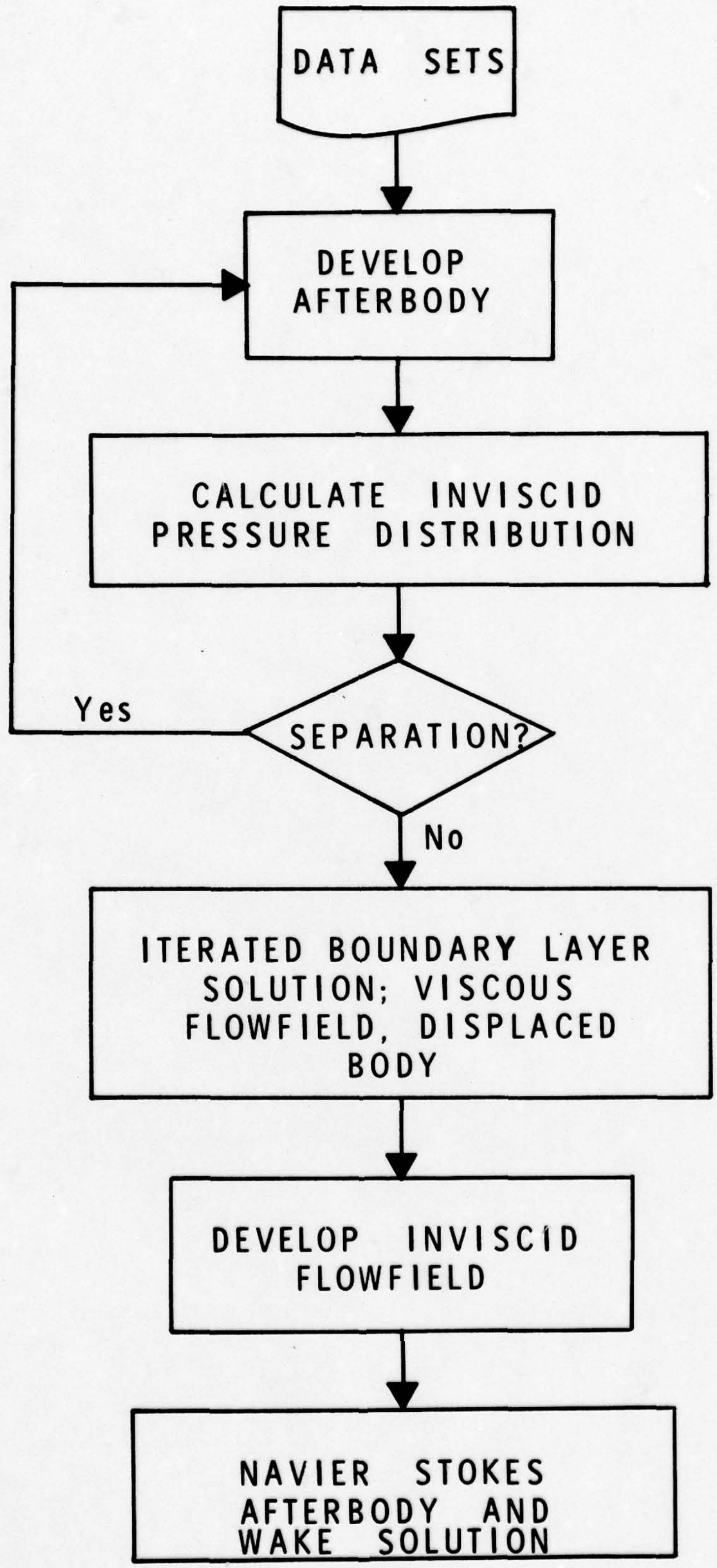


Figure 1. LOGICAL FLOW OF SYSTEM

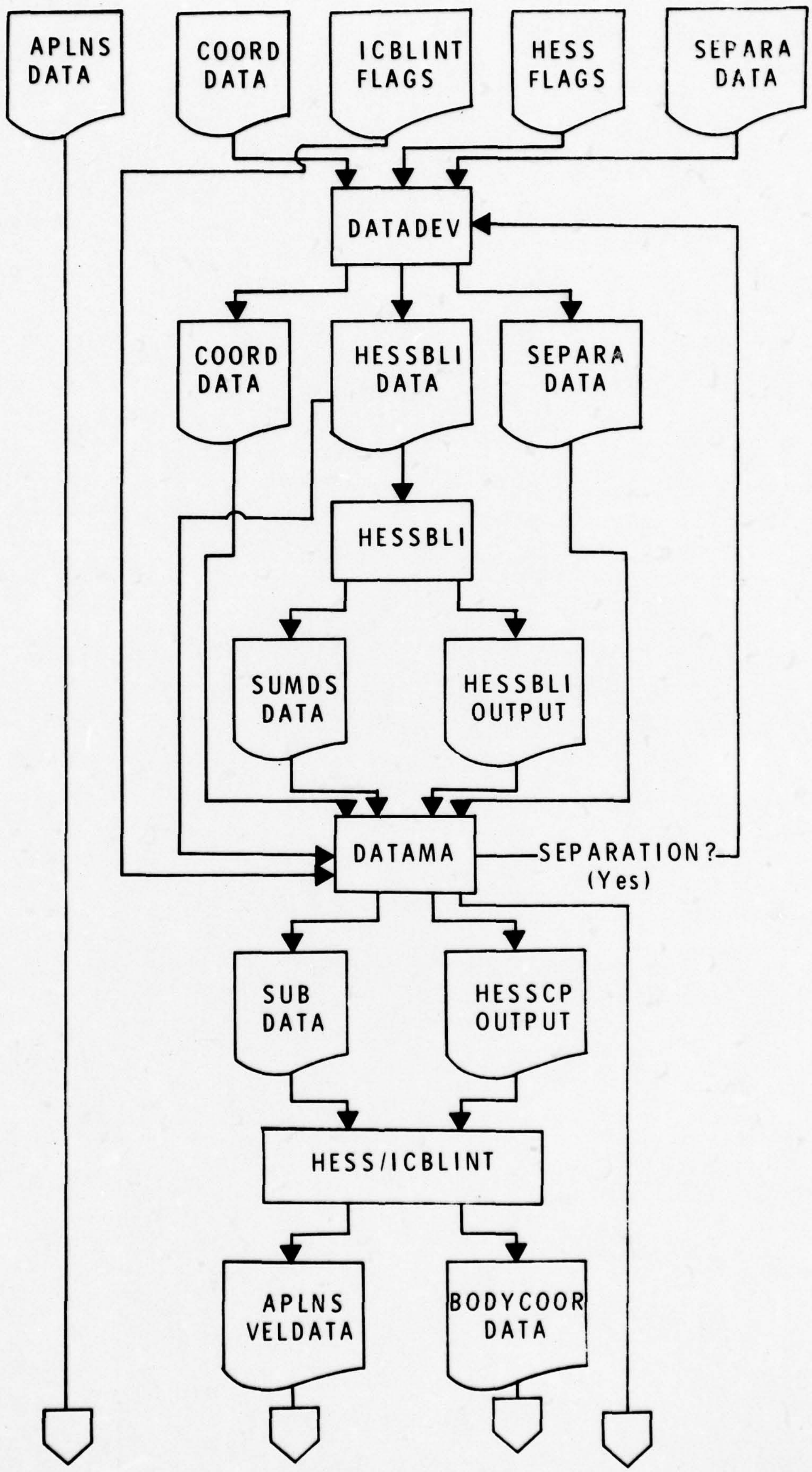


Figure 2. MECHANICAL FLOW OF SYSTEM

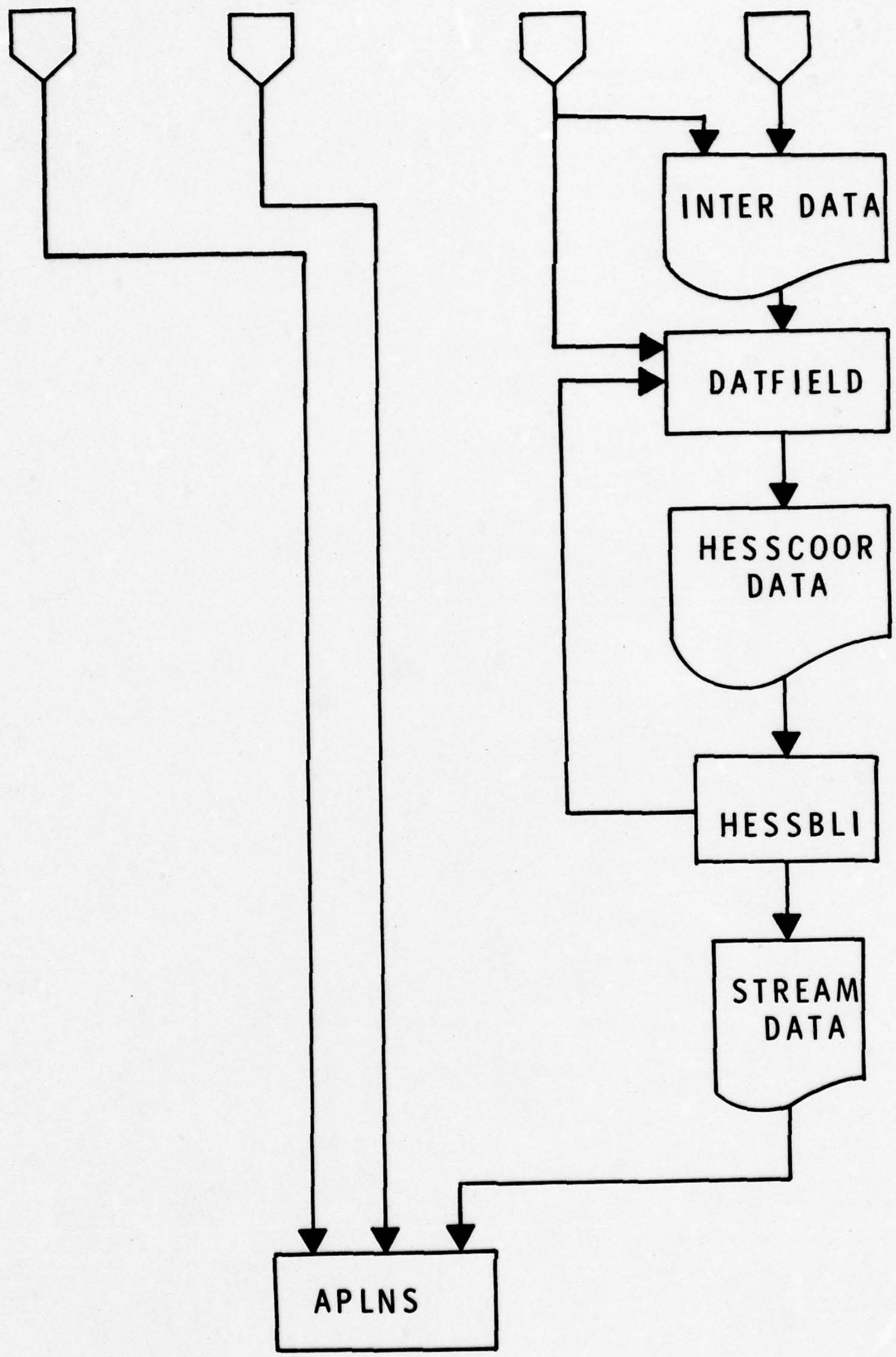


Figure 2. MECHANICAL FLOW OF SYSTEM (continued)

DESCRIPTION OF PROCEDURE CDCP

CDCP EXEC is an exec procedure which ties together the operation of the support programs: DATADEV, DATAMA, and HESS.

Programs DATADEV and DATAMA, controlled by CDCP EXEC, generate a complete set of data files with which the iterative procedure can be executed. The procedure includes the development and the checking of body geometries until a nonseparating flow is found. The development of the nonseparating flow involves: (1) the incremental extension of a trailing edge fairing into the wake (performed in DATADEV), (2) the testing of that fairing in HESS, and (3) the check of that fairing in DATAMA. CDCP EXEC first executes DATADEV, then HESSBLI and then DATAMA execs. When DATAMA has completed executing, the data file SEPARA DATA is checked and the flag ICP is evaluated. IF ICP is 0, a successful afterbody has been found; if 1, the iterative procedure is continued. If a nonseparating fairing has been found, data sets are written and control shifts to BLINT EXEC, but if not, the fairing is extended further into the wake, and the result is retested. This sequence is continued until a successful afterbody is generated or an iteration limit is reached. Because of imperfections in the fitting of the faired afterbody to the original body, as stated above, it is suggested that the user carefully examine the coordinate sets and C_p distribution that are output by CDCP and smooth where needed, especially in the trailing edge region and near the body/fairing joint.

Use of CDCP EXEC (ARGUMENT)

(ARGUMENT) is the maximum number of iterations that the sequence

will complete to find a satisfactory fairing. This is put in as a safety feature to prevent execution from continuing indefinitely due to some unforeseen body characteristic. A listing of CDCP EXEC can be found in Appendix A, Table 2.1.

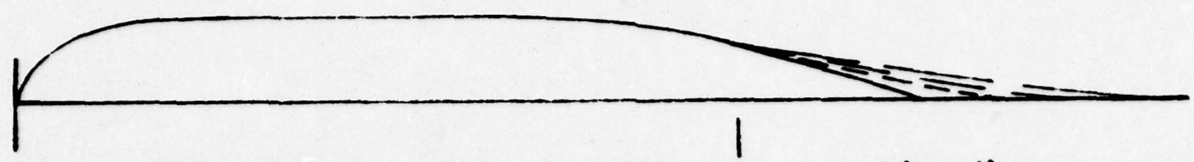
Program DATADEV

Description of Program DATADEV. DATADEV EXEC is the EXEC program which executes DATADEV FORTRAN. DATADEV FORTRAN generates data sets for the HESS potential flow code. On the first pass DATADEV simply writes out a full HESS data set using the original body coordinates. On successive passes, DATADEV generates the afterbody which extends into the wake and further up the body. The extension up the body has been found to hasten afterbody development. The fairing is concave upward, beginning at the point where C_p changes sign from negative to positive (see Fig. 3). After developing the afterbody, new HESS data sets are written out, and a new data set with iteration information is written. A flow chart of DATADEV is found in Fig. 4 .

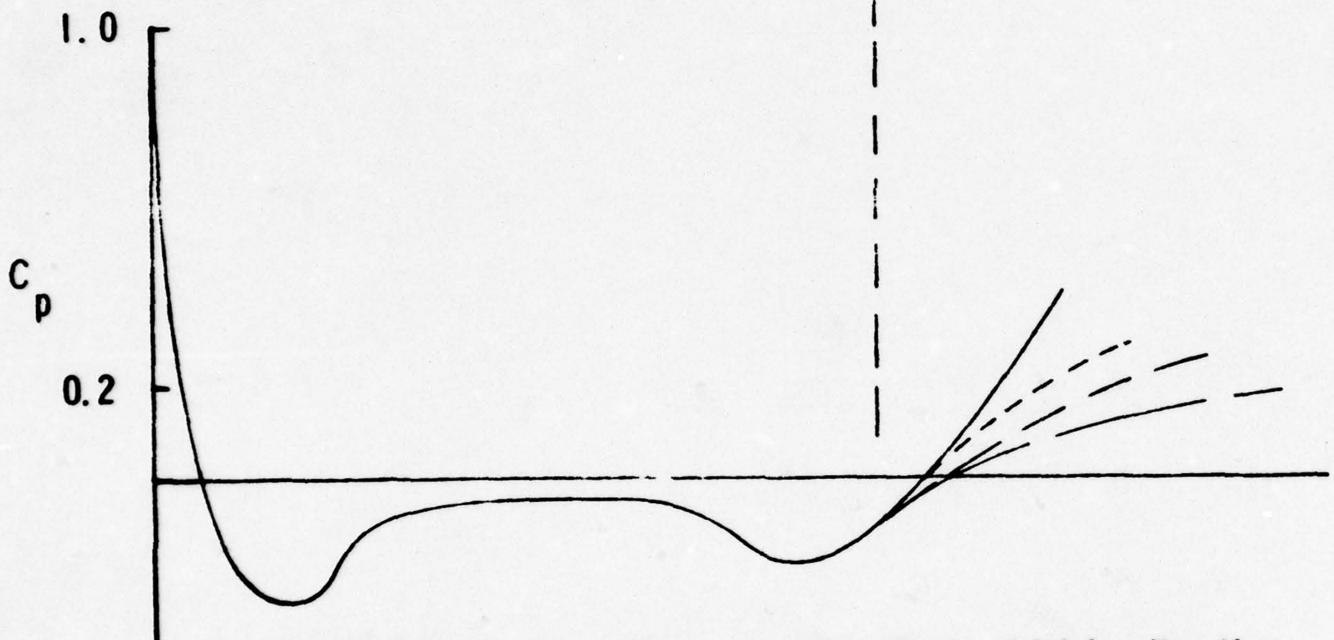
See Appendix A, Tables 2.2 and 2.3 for listings of DATADEV EXEC and DATAMA FORTRAN.

Description of Variables in Common.

BLOCK	VARIABLE	DESCRIPTION OF USE
CORFIT	C	Slope of body at the beginning of the Navier-Stokes region.
CORFIT	DX	Constant stepsize used in re-positioning afterbody coordinates.



Development of fairing extension over several iterations



C_p distribution over the body showing the effects of fairing iteration

Fig. 3 Development and effect of wake fairing extension

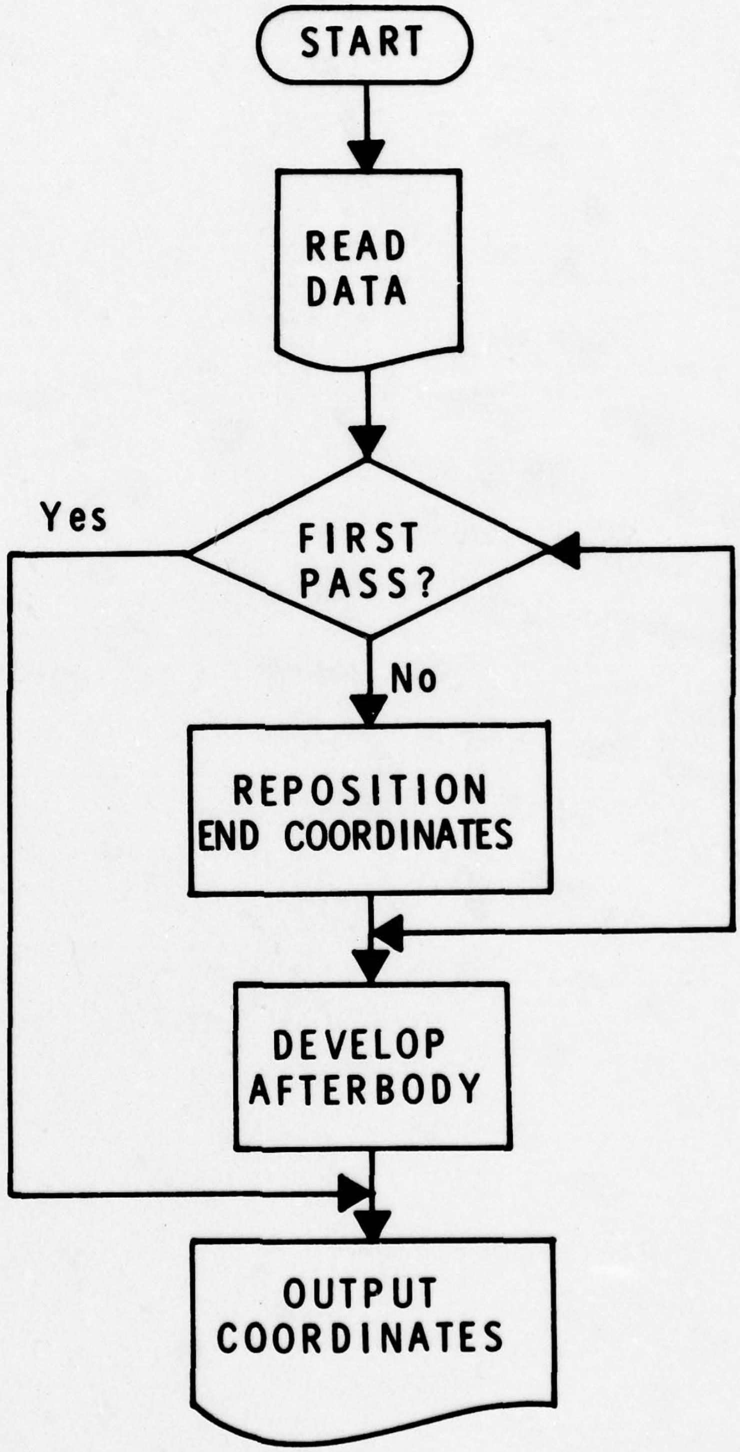


Figure 4. LOGICAL FLOW OF DATADEV

COORD	KTE	Station number of trailing edge on original body.
COORD	KTE1	Station number of trailing edge of faired (extended) body.
CORFIT	NNN	Station number of beginning of re-positioned coordinates.
COORD	NOPRFS	Number of profiles in the part of the Navier-Stokes region on the body.
COORD	X	Original x coordinates on first pass, re-positioned coordinates on subsequent passes.
COORD	Y	Original y coordinates on first pass, re-positioned coordinates on subsequent passes.
CORFIT	X1	Axial distance from the beginning of the fairing to KTE1.
CORFIT	Y1	Radial distance from the beginning of the fairing to the axis.
COORD	XI	Array of x coordinates with afterbody fairing.
COORD	YI	Array of y coordinates with afterbody fairing.

Description of Subroutines.

MAIN. MAIN reads in the body coordinates. On the first pass, the original coordinates are read. On successive passes the original

coordinates with the coordinates of the afterbody repositioned to give a constant x stepsize are read. MAIN then calls LINDEV and HESSPA to generate and write out new coordinates with afterbody.

The variables listed below are used in MAIN. A description can be found in VARIABLES IN COMMON: KWAKE, KTE, KTE1, X, Y.

Subroutine LINDEV. LINDEV develops the trial afterbody fairings and reads in the separation information, including the iteration number. The routine then repositions the coordinates near the trailing edge to give a constant stepsize and on subsequent iterations, extends the fairing into the wake LWAKE points and steps back the beginning of the fairing IWAKE points. This set of x-coordinates is used in developing the fairing which is done in FIT.

Description of Variables.

DX	X stepsize in afterbody for repositioned coordinates. Set at one percent of body length.
KK	Number of repositioned coordinates.
LWAKE	Number of points in wake used in extension of afterbody, LWAKE = IWAKE*5
MM	Number of body stations with repositioned coordinates.

Subroutine FIT. Subroutine FIT generates a curve to fit the boundary conditions given by LINDEV in the form of X1, X2, Y1, and C. C is the body slope at X1, Y1; the slope at the trailing edge is assumed to be zero. The fit is based on the equations.

$$Y'_N = [\Delta Y(K) + \Delta Y(K^2) + \Delta Y(K^3) + \ldots + \Delta Y(K^N)],$$

$$N = 1, KK, KK = \text{No. of points on curve} \qquad (1)$$

$$Y_N = Y_1 - Y'_N \qquad (2)$$

$$K = \frac{\Delta X}{\Delta Y} C \qquad (3)$$

DELY is initially set to 0.0 and successively increased by DY. K is calculated according to Eq. 3. If Y from Eqs. 1 and 2 is less than Y1, DELY is increased and the calculation is redone until the convergence criterion (YK-Y1) = 0.0001 is attained. The Y's based on the last K are used in the afterbody coordinate sets.

Description of Variables.

DELX	X stepsize (= dx)
DELY	Y stepsize
K	Constant multiplier in numerical curve fit
L	Station number of the beginning of the afterbody
Y	Working array of Y coordinates of afterbody

Other variables used in FIT and described in Variables in Common are C, Y1, X1, YI, and DX.

Subroutine HESSPA. HESSPA writes out a complete HESS data set using the body coordinate arrays XI and YI which include the afterbody. The routine first reads in the HESS parameters and includes the changed parameters, if any, in the written set.

Description of Input and Output.

Input Data. Three data sets are input into DATADEV for execution. They are: COORD DATA, SEPARA DATA, and HESS PARMDATA.

COORD DATA contains the original body coordinates together with parameters and flags.

Card 1 KWAKE (I3)

	KWAKE	Flag for including wake points in solution: 0 no wake points 1 use wake points (must have data set of wake points if KWAKE = 1)
Card 2	KTE, NOPRFS, KTE1 (1x,3I3)	
	KTE	Station number of trailing edge on original body (first pass original T. E., successive passes repositioned T. E.)
	NOPRFS	Number of profiles on the body in the APLNS box. Maximum of 15 is the recommended value
	KTE1	Station number of trailing edge on faired body
Cards 3-KTE+3	X,Y	(2F12.6)
	X	x coordinate of the original body geometry
	Y	y coordinate of the original body geometry

See Appendix B, Table 1.1 for a sample COORD DATA file.

SEPARA DATA contains information on the status of the iterations and flags on separation.

Card 1	IWAKE	Iteration counter, on first iteration has a value of 0
Card 2	JSTA, KTE	(1x,3I3)
	JSTA	Station number of start of faired body. For initial data set can be 0

	KTE	Described above
Card 3	ICP	(I3)
	ICP	Separation flag, 0 for no separation; 1 for separation. Set to 0 initially.

See Appendix B, Table 1.2 for a sample SEPARA DATA data set.

HESS PARMDATA contains the HESS execution parameters. A further description can be found in the HESS user's guide.

Card 1	HEDR, CASE	(15A4,2x,2A4)
	HEDR	Alphanumeric array with the case title
	CASE	Alphanumeric array with the case number
Card 2	ICARD2	(I3)
	ICARD2	HESS execution flags, for body only solution set at 1,0,1
Card 3	ICARD3	(I5)
	ICARD3	HESS execution flags, set to 0 (see HESS user's guide)
Card 4	ICARD4, NN	(1x,I4,I5)
	ICARD4	HESS execution flags, for body only solution set to 1,0,0,1
	NN	Number of body stations. Set to KTE initially and will be changed to KTE1 for separating case
Card 5	IBDN	(9x,I1)
	IBDN	Body number, for body only solution set to 1

See Appendix B, Table 1.3 for a sample HESS PARMDATA data set.

Output Data. The output files created by DATADEV are: SEPARA DATA, COORD DATA, and HESSBLI DATA.

SEPARA DATA	Described above. Rewritten for use by DATAMA
COORD DATA	Described above. Values of KTE1 and original coordinates changed to reflect fairing extension (KTE1) and repositioning of original body coordinates
HESSBLI DATA	Complete data set for HESS execution, including fairing when one is used (second and subsequent iterations)
Card 1-5	Described above in HESS PARMDATA
Card 6 - KTE1 XI, YI	(2F12.6)
XI,YI	Body coordinates

See Appendix B, Table 1.4 for a sample HESSBLI DATA data set.

Program Datama

Description of Program Datama. DATAMA EXEC executes program DATAMA. DATAMA evaluates the pressure distribution developed by HESS and decides if the geometry will produce separating flow. The program also interpolates the C_p data set at on-coordinate points for use in the HESS/ICBLINT iterative scheme. If the developed afterbody successfully prevents separation, the full HESS/ICBLINT data sets are written. An option is included to add wake points for a wake solution from HESS/ICBLINT. A flow chart of DATAMA is shown in Fig. 5 .

See Appendix A, Tables 2.4 and 2.5 for listings of DATAMA EXEC and DATAMA FORTRAN.

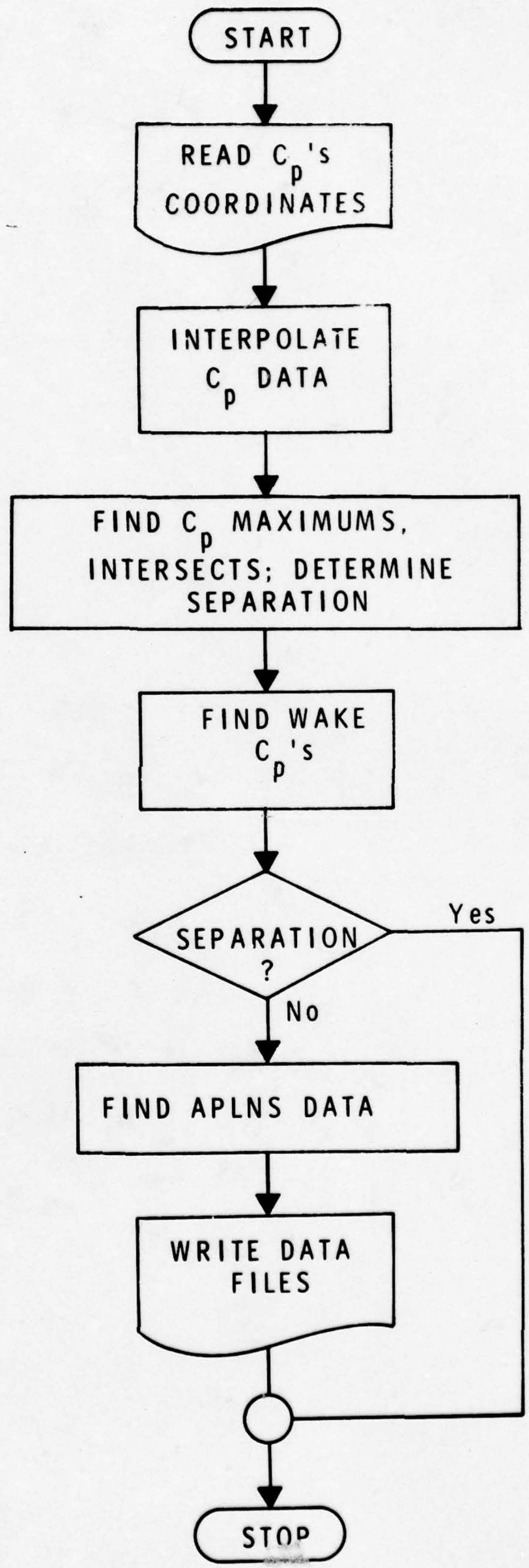


Figure 5. LOGICAL FLOW OF DATAMA

Description of Variables in Common.

Variable	Description of Variable
DYDX	Slope of straight taper approximation of tail section (without afterbody) for use in APLNS geometry data
JSTA	Station near trailing edge at which the C_p changes sign (- to +), which determines the beginning of the fairing extension.
ICP	Separation flag; 1 if flow is separating, 0 if otherwise
IIMAX	Maximum number of coordinate stations. Values in coordinate array including wake points
JWAKE	Number of wake points at which a solution is to be found
KTE	Station number of original body
KTE1	Station number of trailing edge of body with fairing
KWAKE	Wake flag; 1 if wake points are to be used, 0 if otherwise
X	Array of faired body coordinates
XI	Array of original body coordinates
XLOCAT	X coord of beginning of fairing X(JSTA)
Y	Array of final y body coordinates with fairing
YI	Array of original body coordinates

YWAKE	Array of Y values of wake coordinates.

Description of Subroutines.

MAIN. MAIN reads in wake coordinates, if used, and calls subroutines CPDATA, CURVE, and READAT. The last two routines are called only if ICP = 0.

Subroutine CPDATA. Subroutine CPDATA is the working section of the program. Pressure coefficients are read in from the previous HESS execution and, since these coefficients are given at off-coordinate points, the data are interpolated to on-body coordinate points. These C_p data are examined to find locations of sign changes, negative minimums and magnitudes over the fairing. If the fairing C_p data go higher than 0.2, the body is assumed to produce separating flow. If wake points are to be used, they are added to the coordinate array and given a constant C_p value of the last negative minimum.

Description of Variables.

CP	Array of interpolated C_p values
F	Array of C_p values for off-body points
ICPAR	Station number of negative minimum C_p used in giving initial values to wake C_p data
IWAKE	Iteration counter
LWAKE	Number of points in wake used in fairing extension = IWAKE*5

Subroutine CURVE. CURVE is called when ICP = 0. CURVE reads in the original body coordinates and finds the slope of the body at XI(KSTA). The procedure used defines the straight tapered tail for

the APLNS geometry. CURVE also prints the necessary data for DEVELOP to calculate the inviscid, perturbed flowfield.

Subroutine READAT. Subroutine READAT reads in the HESS/ICBLINT execution parameters and writes out the complete working data sets. For coordinates the routine uses the last (now nonseparating) faired set developed in DATADEV found in HESSBLI DATA. The routine also writes out the stations at which complete solutions in ICBL are to be obtained (IPR array) and the stations at which solutions are to be written out (IPRINT array). For simplicity these arrays are equal, the stations used are dependent on IIMAX.

Description of Variables.

K	Station number at which IPRINT array begins to be incremented by 1.

Description of Input and Output.

Input Data. There are seven data sets used by DATAMA: WAKE COORDATA, ICBLINT PARMDATA, HESSBLI DATA, SEPARA DATA, SUMDS DATA, HESSBLI OUTPUT and COORD DATA. Of these data sets, HESSBLI DATA, SEPARA DATA, and COORD DATA have been described above in INPUT and OUTPUT for DATADEV.

WAKE COORDATA. WAKE COORDATA contains the coordinates of wake points if a wake solution is desired. The input data are as follows:

Card 1	JWAKE	(I3)
	JWAKE	JWAKE is the number of wake points in the arrays XWAKE and YWAKE
Card 2-KWAKE+1	XWAKE	(F12.6) KWAKE is the number of points in wake coordinate data set

XWAKE — The array of values of coordinates in the array of wake points.

See Appendix B, Table 1.5 for a sample WAKE COORDATA data set.

ICBLINT PARMDATA. ICBLINT PARMDATA is the data set of HESS/ICBL execution parameters. The data set contains all the necessary ICBLINT/HESS data except for the axial, radial, and surface coordinates. The data are updated in DATAMA to include any changes made in the coordinates. For a complete description of ICBLINT PARMDATA see the ICBLINT user's guide, "Description of Input," under SUB DATA. SUB DATA contains in Cards 1-20 the set ICBLINT PARMDATA.

SUMDS DATA. SUMDS DATA is the data set output by HESS giving the surface distance to each of the coordinates which becomes the XSTA array used in ICBLINT. The data set is read in by READAT and written on SUB DATA.

Card 1-NN	XSTA	(12x,F12.6)

See Appendix B, Table 1.6 for a sample SUMDS DATA data set.

HESSBLI OUTPUT. HESSBLI OUTPUT is the off-coordinate C_p data from HESS.

Cards 1-NN XW ← XTO ← F(3F12.6)

XW	XW is the off-coordinate point at which the C_p data have been calculated in HESS
XTO	XTO is the on-coordinate point to which the C_p data will be interpolated
F	F is the off-coordinate C_p value

Output Data. DATAMA outputs five data files: SUB DATA, HESSCP OUTPUT, SEPARA DATA, DATAMAKE DATA, and INTER DATA.

SUB DATA. SUB DATA is the complete data file for HESS/ICBL execution with all parameters and coordinate points (see INPUT FOR ICBLINT).

HESSCP OUTPUT. HESSCP OUTPUT is the final, interpolated data set of C_p values for use in HESS/ICBL along with SUB DATA. The data contain the on-coordinate C_p values. HESSCP OUTPUT is described in the ICBLINT user's guide, "Description of Input."

See Appendix B, Table 1.7 for a sample HESSCP OUTPUT data set.

SEPARA DATA. SEPARA DATA, described earlier, is changed in DATAMA, and output again with the new information.

DATAMAKE DATA. DATAMAKE DATA is the data set for execution of BLDMKER FORTRAN and is simply the number of points in the final body coordinate set.

Card 1	NN	(I3)
	NN	NN is the number of body points

See Appendix B, Table 1.8 for a sample DATAMAKE DATA data set.

INTER DATA. INTER DATA is the data set for DATFIELD to use in developing the inviscid flowfield.

Card 1	IHESSI	(5I1.I5)
	IHESSI	Counter for the number of repetitions completed through DEVELOP
Card 2	XLOCAT,DX,DY	(3F12.6)
	XLOCAT	Location of the upstream edge of Navier-Stokes region
	DX	x stepsize for placement of off-body points

DY y stepsize for placement of off-body points

See Appendix B, Table 1.9 for a sample INTER DATA data set.

DESCRIPTION OF PROCEDURE BLINT

BLINT EXEC is the controlling exec procedure for the HESS/ICBLINT system of codes. The iterative scheme involves the execution of these programs, together with the support program BLDMKER in a loop. The number of loops is user determined, since an accurate criterion for convergence is difficult to define. In using the method, it has been found that generally convergence is obtained after the sixth iteration. For simple geometries, fewer than six iterations are required.

The HESS/ICBLINT iterative sequence is the prime data developer of the APLNS program. In the present form the sequence uses the HESS potential flow program in very nearly its original form, the exceptions being the output of off-body point solutions, pressure distributions, and surface coordinates. Program ICBLINT has been modified to both accept and employ in calculations, C_p values from the inviscid solution, calculation of wake properties and a restart capability. All of these are used in the viscous/inviscid, iterated solution. The concepts and theory of this iterative procedure have been described, and the following is a description of the code mechanics.

Figures 6 and 7 show the operation of BLINT. The procedure begins with the execution of ICBLINT to yield the initial displaced body BODYCOOR DATA, which is run by HESS. HESS output is the C_p distribution on the body. The C_p data set (HESSBLI OUTPUT) is manipulated by BLDMKER to yield HESSCP OUTPUT which, together with the original coordinates and the displaced body coordinates, forms the input for the second iteration. In the second and subsequent iterations, the original body and the displaced body coordinates are averaged together (weighting factor of 0.5)

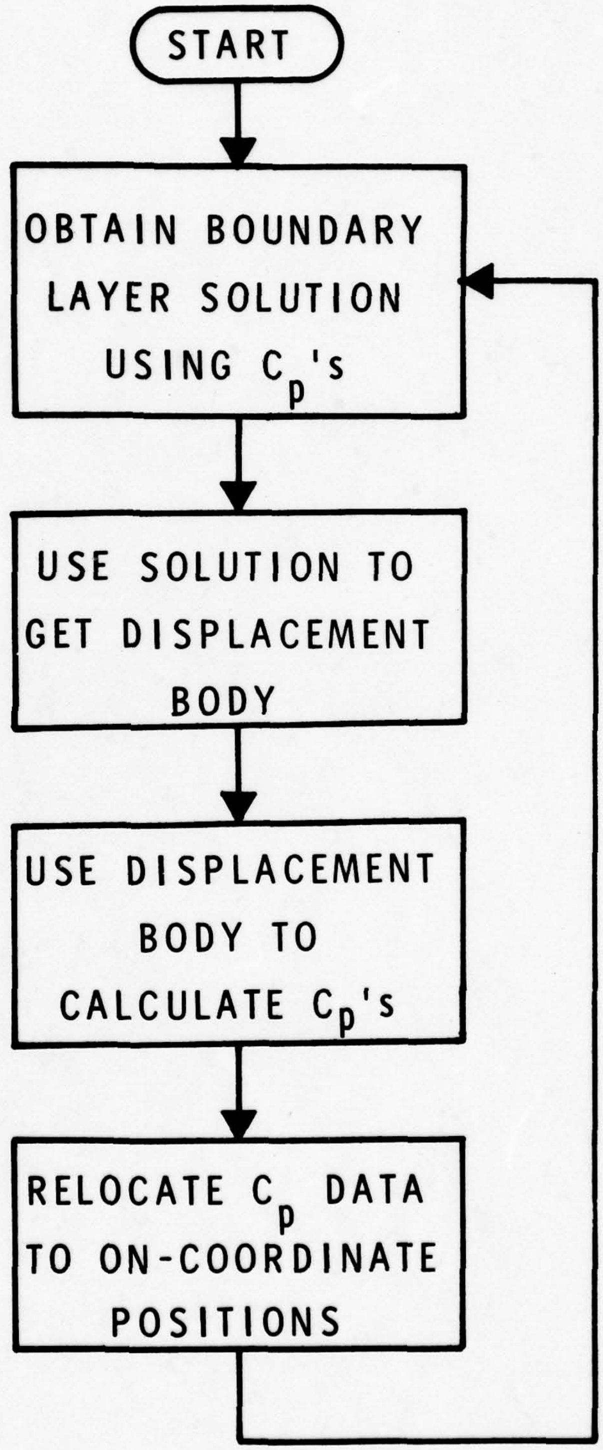


Figure 6. LOGICAL FLOW OF HESS/ICBLINT

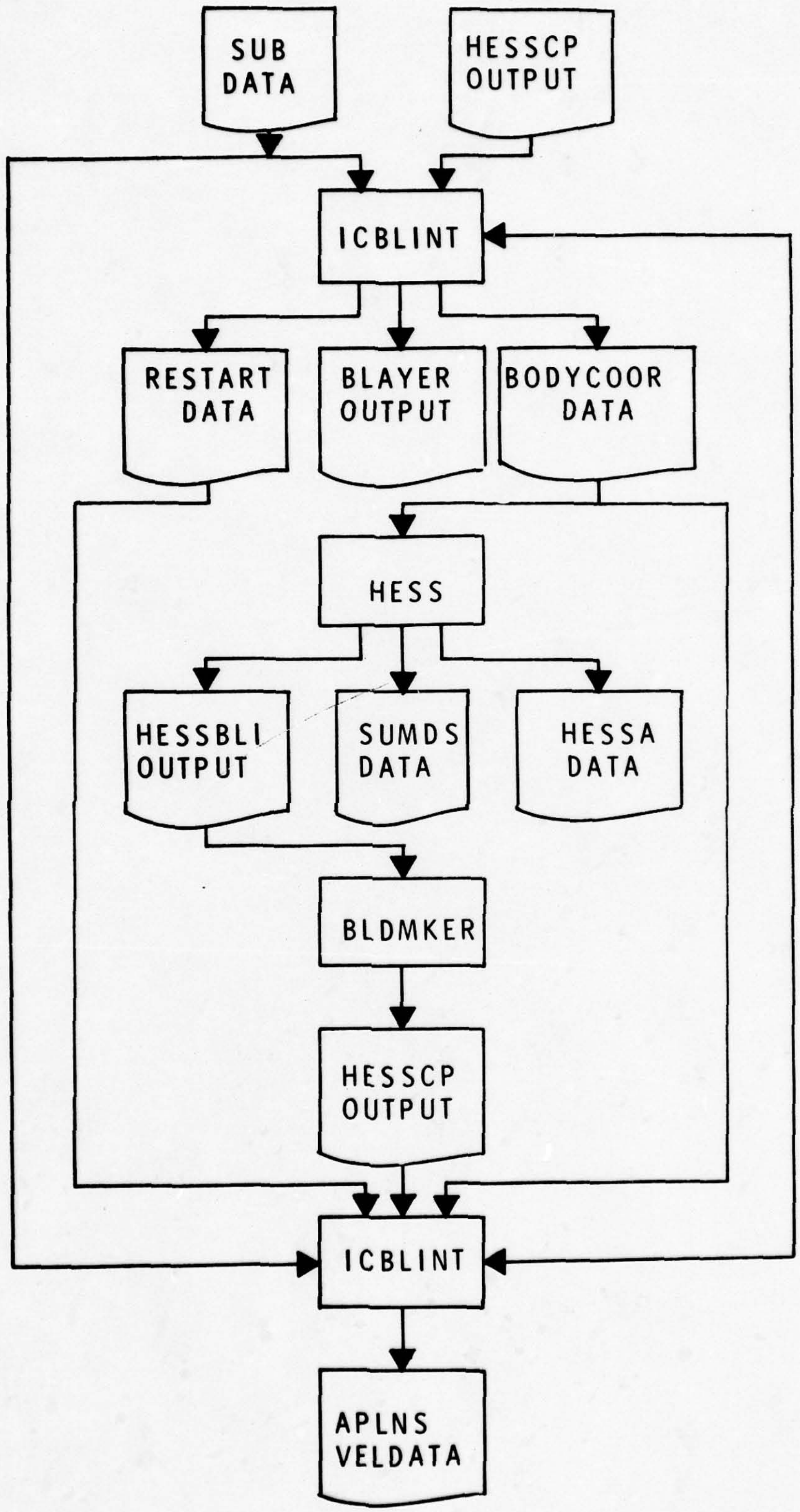


Figure 7. MECHANICAL FLOW OF HESS/ICBLINT

in ICBLINT. Each run yields a displaced body (again BODYCOOR DATA) which is used as HESS input. The procedure then repeats itself for the number of iterations desired.

The procedure for the wake calculations enables the user to obtain iterated wake solutions for nonseparating, no-propeller cases. The extension into the wake may be as far as desired within the array dimensions which will be discussed later. Data set WAKE COORDATA containing the wake points has been described earlier.

The restart capability is a method of saving execution time on the second and subsequent executions of ICBLINT. There is virtually no interaction on the forward region of most bodies and therefore no reason to redo boundary-layer calculations in this region. Therefore, a tape is written at the restart station, and in subsequent iterations the tape is read and the solution begun at the restart station. The actual location of this station is dependent on geometry and must be determined initially by the user.

See Appendix A, Table 3.1 for a listing of BLINT EXEC.

Program HESS

The HESS potential flow code is controlled by HESSBLI EXEC. Only three modifications have been made to the original version of the Douglas computer program and are as follows:

1. The pressure coefficients are written out on a separate tape (HESSBLI OUTPUT).
2. The complete inviscid, off-body flowfield is written on a separate tape, and added on during each execution. Details of this modification are found in the description of program

DEVELOP, where the modification is used.

3. The surface distance from the leading edge (SUMDS DATA) is written out on separate tape for use in ICBLINT input data sets.

The complete HESS user's guide is sent under separate cover, describing the code and is accurate except for the above modifications. A listing of HESSBLI EXEC is given in Table 3.2 of Appendix A.

Program BLDMKER

Description of Program BLDMKER. BLDMKER is an interpolating program which performs the same function as Program DATAMA in routine CPDATA. BLDMKER interpolates the C_p data output by HESS in HESSBLI OUTPUT and finds the C_p at on-coordinate positions, creating HESSCP OUTPUT.

See Appendix A, Table 3.3 and Table 3.4 for listings of BLDMKER EXEC and BLDMKER FORTRAN.

Description of Input for BLDMKER. BLDMKER requires two input data files: DATAMAKE DATA and HESSBLI OUTPUT.

DATAMAKE DATA is the number of points on the body.

Card 1	NN	(I3)
	NN	Number of body points.

HESSBLI OUTPUT has been described in "Input of DATAMA."

Description of Output for BLDMKER. This is described in "Input to Program ICPL."

Program ICBLINT

Description of the Program ICBLINT. Program ICBLINT has been developed to provide solutions to a large number of incompressible boundary-layer flows. The program can calculate either body only or body

and near wake cases and can treat 2-D or axisymmetric geometries with blunt or sharp noses. Solutions may be obtained using only ICBLINT or ICBLINT in a viscous-inviscid iterative procedure with pressure coefficients of the displaced body as the medium of iteration. The program can predict laminar and/or turbulent flows, the transition models used are described in the contractor report under separate cover.

The flow of logic in the program is controlled by MAIN (see description of MAIN). To assist the user, two flow diagrams of MAIN are included. Figure 8 gives a flow chart of MAIN by the logical functions used, while Fig. 9 gives a flow diagram of MAIN by the subroutines called. As the major subroutines have particular logical functions, there is a close correlation between the two flow charts.

Additional Comments on the use of HESS/ICBLINT. Although the number of body points allowed is a maximum of 501, for the body problems of concern here, fewer than 101 points are preferred. The reasons for this are twofold: first, only a maximum 101 profiles may be solved, the remainder will be used for body definition only; and second, the solutions at only 101 points may be written and hence used in the development of the displaced body in the Potential Flow Code and for the displaced body definition in ICBLINT. For separating flows, the number of body points should be checked and, if necessary, adjusted after CDCP EXEC has been executed.

See Appendix A, Table 3.5 for a listing of BLAYER EXEC and Table 3.6 for a listing of ICBLINT FORTRAN.

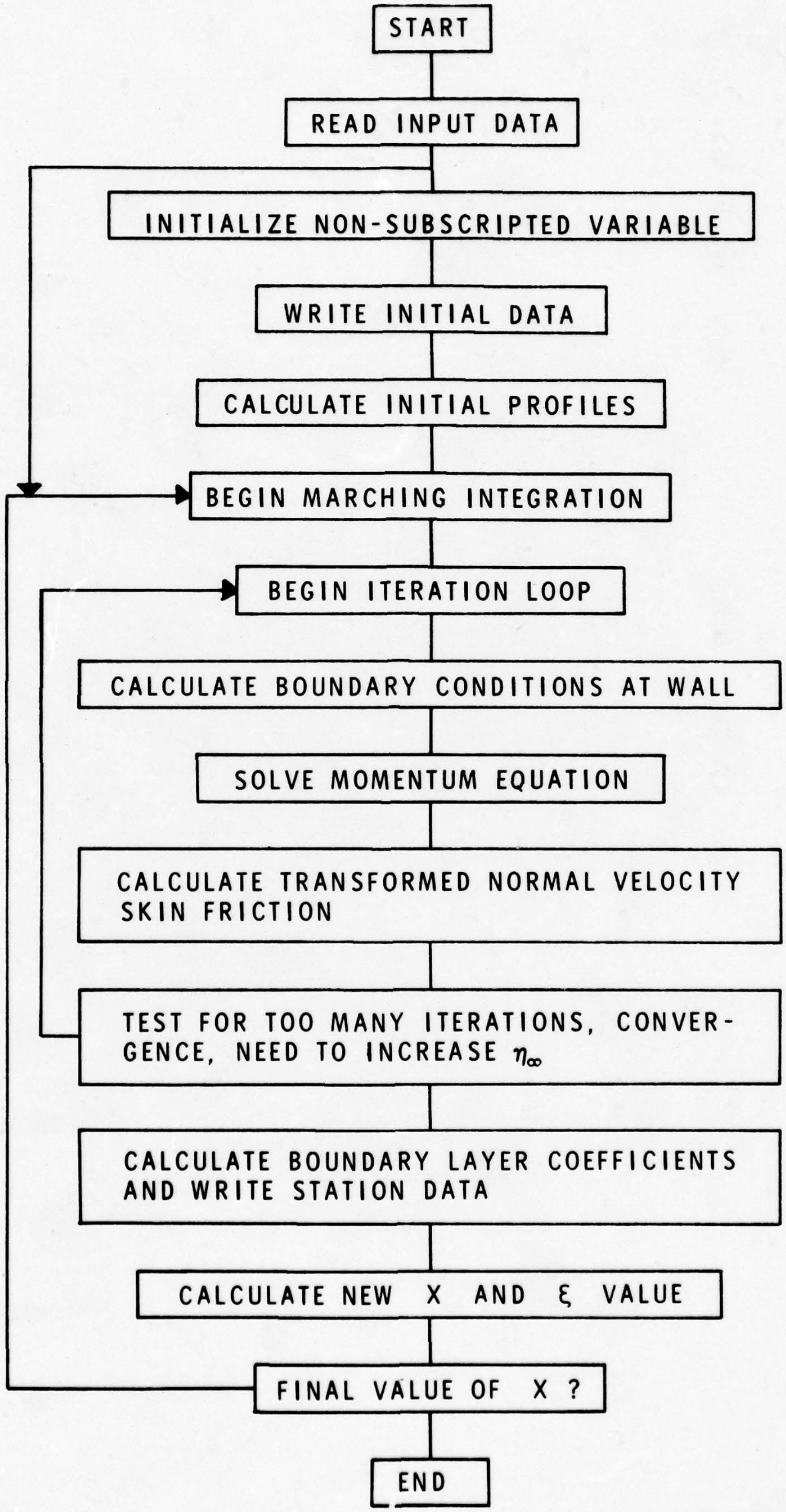


Figure 8. FLOW DIAGRAM OF MAIN ROUTINE BY FUNCTION

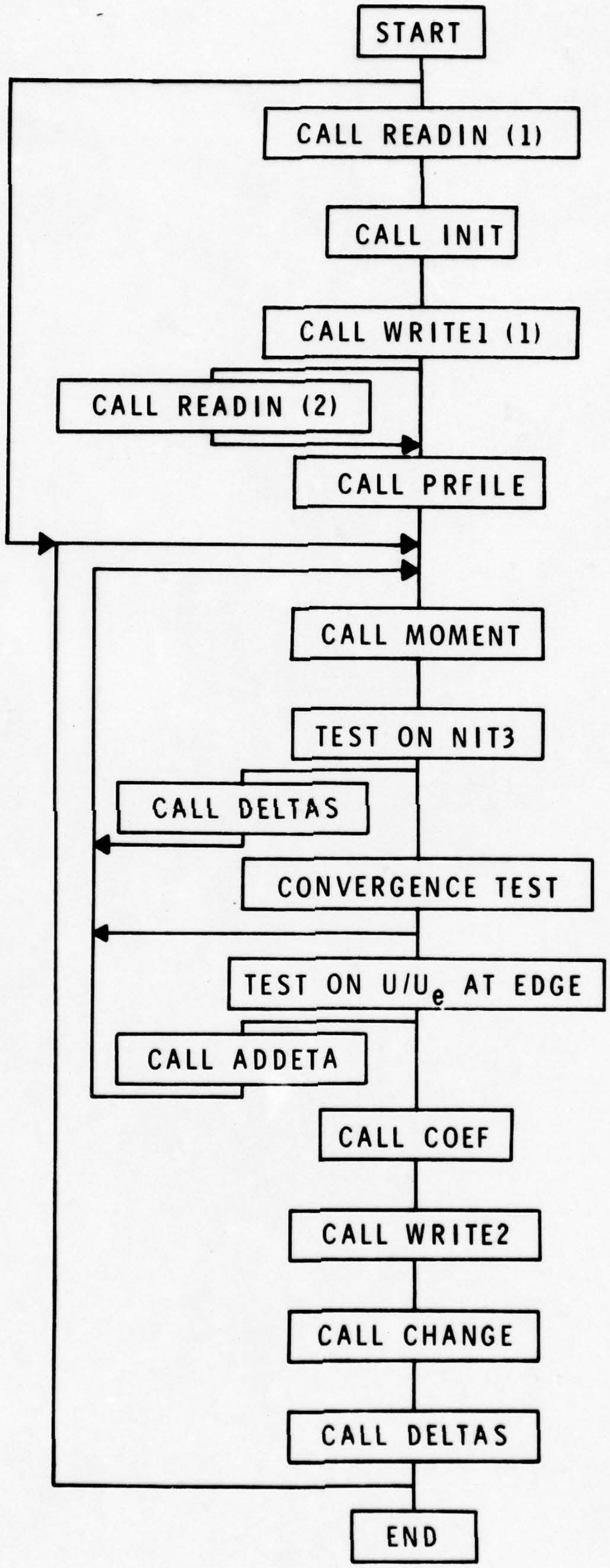


Figure 9. FLOW DIAGRAM OF MAIN ROUTINE BY SUBROUTINES CALLED

Description of Variables in Common. In this section, each variable which is included in a labeled common block is given a brief description. The name of the labeled common block in which the variable occurs is given to the left of the variable name.

Common Block	Variable Name		Description of Variable
NMLCRD	ADTEST	,	Criterion for Increasing ETAINF, (F(IE) - F(IE-4) is compared with ADTEST)
GEOME	ANGLE	,	Local Body Angle, in Degrees (Converted to Radians)
TRANS	ATR	,	Transition Constant (=0.412)
ARRAY1	AOB	=	A_0
ARRAY1	AOBP	=	$\frac{\partial A_0}{\partial \eta}$
ARRAY1	A1	,	Partial Differential Equation Coefficient
COMWLL	A1B	,	Derivative of U/U_e at Wall
ARRAY1	A2	,	Partial Differential Equation Coefficient
ARRAY1	A3	,	Partial Differential Equation Coefficient
ARRAY1	A4	,	Partial Differential Equation Coefficient
EDGPRP	BETA	=	$\beta = 2\xi/U_e(dU_e/d\xi)$
CNVERG	CCRNI	=	1 - CRNI
CFPR	CF	,	Skin Friction in Transformed Coordinates ($= C_{f_\infty}/\varepsilon_{VD}$)

COEFF	CFBAR	=	$\bar{C}_f = \frac{1}{x}\int_0^x C_{f_\infty}\, dt$
COEFF	CFBREX	=	$\bar{C}_f \sqrt{REX}$
COEFF	CFE	=	$C_{f_e} = 2\tau_w/\rho_e U_e^2$
COEFF	CFINF	=	$C_{f_\infty} = 2\tau_w/\rho_\infty U_\infty^2$
COEFF	CFRES	=	CFREY
COEFF	CFREY	=	$C_{f_\infty} \sqrt{REX}$
COEFF3	CF1	,	Term used in Computing CFBREX
ARRAY1	CHI	=	$\chi = \frac{y^2}{\nu}\frac{\partial u}{\partial y}$, Vorticity Reynolds Number
TRANS	CHICRT	,	Value of CHIMAX at which Transition Regime Calculations are Begun
TRANS	CHIMAX	,	Maximum Value in CHI Array
CNVERG	CONVRG	,	Convergence Criterion
CNVERG	CRNI	,	Selector for Finite Difference Scheme; 1.0 for Fully Implicit, 0.5 for Crank-Nicolson, and 0.0 for Fully Explicit
COEFF	DEL	=	δ, Boundary Layer Thickness (Value of y where $U/U_e = 0.995$)
COEFF2	DELORF	=	δ/reference length
COEFF2	DELOX	=	δ/x
COEFF	DELST	=	δ^*, Boundary Layer Displacement Thickness
COEFF	DELTA	=	δ/ϵ_{VD}

CNVERG	DIF	,	Maximum Difference in G or F (or their Derivative at the Wall) between Iterations (Compared with CONVRG in Convergence Test)
CNVERG	DIFF	,	Local Difference in G or F between Iterations
CNVERG	DIF2	,	Differnece in the Derivative of F at the Wall Between Iterations
ARRAY1	DN	,	Array of $\Delta\eta$ Values
ARRAY1	DN2	,	Array of $\Delta\eta$ Values (Used only when ETAINF is Increased)
GEOME	DS	=	$\Delta\xi$ Corresponding to Δx
COEFF2	DSAXOR	=	δ^*_{axi}/reference length
COEFF	DSTARK	=	δ^*_k, Incompressible Boundary Layer Displacement Thickness
COEFF2	DSTODL	=	δ^*/δ
COEFF2	DSTORF	=	δ^*/reference length
COEFF2	DSTOTH	=	δ^*/θ
COEFF2	DSTOX	=	δ^*/x
COEFF	DSTRAX	=	$\delta^*_{axi} = r\,\{\sqrt{1+2\delta^*/r} - 1\}$ (Zero for 2-D Flow)
EDGPRP	DUEDS	=	dUE/ds
GEOME	DX	=	Δx, Step Size in x Direction
GEOME	DXMAX	,	Maximum Value of Δx Permitted in the Calculations
GEOME	DXOLD	,	Previous Value of Δx
GEOME	DX1	,	Storage Value of Δx at the First Station

GEOME	DYDX	,	Body Slope at JSTA
ARRAY1	EPSO	=	ε_0^+, the Outer Eddy Viscosity
ARRAY1	EPSPL	=	ε_i^+, or ε^+, the Inner Eddy Viscosity or the Combined Inner and Outer Eddy Viscosities
VSCSTY	EPSVD	=	$\varepsilon_{VD} = \{\mu_{ref}/(\rho_\infty U_\infty L)\}^{\frac{1}{2}}$, where L is the Unit Length in REINF
NMLCRD	ETAINF	,	Maximum Value of η
COMWLL	E1	,	Matrix Inversion Coefficient Evaluated at the Wall
ARRAY1	FC	=	CRNI(F2-F1) + F1
ARRAY1	FCN	=	CRNI(F2N-F1N) + F1N
ARRAY1	FCP	=	(F2-F1)/DS
COMWLL	FF	,	Wall Boundary Condition on F
ARRAY1	F1	,	Value of F (U/U_e) from Previous Iteration or Station
ARRAY1	F1N	=	$\frac{\partial F1}{\partial \eta}$
ARRAY1	F1NN	=	$\frac{\partial^2 F1}{\partial \eta^2}$
ARRAY1	F2	,	Value of F from Solution of Momentum Equation
ARRAY1	F2N	=	$\frac{\partial F2}{\partial \eta}$
ARRAY1	F2NN	=	$\frac{\partial^2 F2}{\partial \eta^2}$
COMWLL	F2N1	,	Value of F2N at Wall from Previous Iteration

TRANS	GAMMA	,	Transition Intermittency Factor, Γ, $\Gamma = 1 - \exp(-A\bar{\xi}^2)$, where A = ATR and $\bar{\xi}$ = XIBAR
COEFF	HAFCF	=	$C_{f_\infty}/2$
INTGR	IE	,	Number of Grid Points in η Direction
NTEGER	II	,	Subscript for Array IPR
INTGR	IIMAX	,	Maximum Number of Values in Geometry Arrays (i.e. ZA, XSTA, etc.)
INTGR	IM	=	IE - 1
NTEGER	IPFL	,	Number of Values in Array IPRFL
ARRAY2	IPR	,	Array of Subscripts for Array XSTA Giving Values of x at which a Solution is to be Obtained
ARRAY2	IPRFL	,	Array of Subscripts for Array XSTA Giving Values of x at which Complete Solutions (with Profiles) are to be Printed
NTEGER	IPRINT	,	Number of Values in Array IPR
INTGR	IPRNT	,	Subscript for Array IPRFL
INTGR	ISTOP	,	Number of Iterations Since Last Converged Solution
NTEGER	JJ	,	Integer Used in Computing DX
APLDAT	JSTA	,	Beginning of N-S Box
NTEGER	K	,	Station Counter
NTEGER	KADETA	,	Indicator, if Zero, ETAINF is Held Constant, if Non-zero, ETAINF can be Increased

NTEGER	KEND	,	Maximum Value Allowed for K
NTEGER	KEP	,	Subscript in EPSPL Array Where the Inner and Outer Eddy Viscosities are Matched
INTGR	KL	,	Index for Profile Output DO Loop, 1 Gives Output for Each Grid Point, 2 for Every Second Grid Point
NTEGER	KSTOP	,	Indicator, Normally Zero, Set to One when Solution is Completed (X = XSTA (IIMAX))
TRANS	KTRANS	,	Indicator for Transition Regime, Zero for No Transition or Instantaneous Transition, One for Transition Regime (Set at Two at "End of Transition")
NTEGER	KTRNSN	,	Subscript in XSTA Array at Which Instantaneous Transition Occurs
INTGR	KVSLAW	,	Indicator for Inner Eddy Viscosity Law, Zero for Reichardt Law, Non-zero for Van Driest Law
NTEGER	LAMTRB	,	Indicator, 1 for Laminar Flow, 2 for Turbulent Flow
CNVERG	NC	,	Specifies Convergence Test, Zero for Test on Derivatives at Wall, One for Test on Functions at all Points in Profile
INTGR	NIT	,	Number of Iterations at Current Value of x
NTEGER	NITTOT	,	Cumulative Total Number of Iterations in x Direction

INTGR	NIT1	,	If, after a converged solution is obtained, NIT is less than or equal to NIT1, Δx is doubled
INTGR	NIT2	,	If, after a converged solution is obtained, NIT is greater than or equal to NIT2, Δx is halved
INTGR	NIT3	,	If NIT = NIT3 before a converged solution is obtained, Δx is halved, X and XI are recomputed and the iteration loop is restarted with the smaller values of X and XI
APLDAT	NOPRFS	,	Number of Profiles in N-S Box
INTGR	NOSE	,	Indicator of Geometry, 1 for a Blunt Nose, 2 for a Sharp Nose
STRT	NRSTRT	,	Increment in K for which the Restart Tape is Written
EDGPRP	PE	=	P_e/P_{ref}, where $P_{ref} = \rho_{ref} U^2_{ref}$
EDGPRP	PESO	=	P_e/P_{ref}, at s = 0
FRSTRM	PFS	,	Freestream Static Pressure, PS A
GEOME	PNC	=	$\frac{\sqrt{2\xi}}{u_e \; r_w^j}$, Term in Transformation of Normal Coordinate
EDGPRP	PP	=	dPE/ds
REF	PREF	=	$P_{ref} = \rho_{ref} U^2_{ref}$
STAG	PSTAG	,	Freestream Stagnation Pressure, PSFA

ARRAY2	PZ	,	Local Pressure Array, P/P_{ref}
STAG	P1O	=	P_0/P_{ref}
GEOME	REFLEN	,	Reference Length (may be any Value Appropriate to User's Needs
FRSTRM	REINF	,	Freestream Reynolds Number per Unit Length
COEFF	RETHET	=	Re_θ, Edge Unit Reynolds Number Times θ
COEFF	REX	,	Local Reynolds Number Based on Edge Conditions
FRSTRM	RHOFS	=	ρ_∞, slugs/ft^3
REF	RHOREF	=	$\rho_{ref} = \rho_\infty$, slugs/ft^3
COEFF2	ROREFL	=	r/reference length
ARRAY2	RZ	=	r, Array of Radius Values
GEOME	RO	=	r, Local Radius
GEOME	SCF	,	Scale Factor used to Convert Unit Length for REINF (REINF is Computed per Foot if SCF=1; per inch if SCF=12.)
COEFF3	SUM	,	Term used in Computing CFBREX
COEFF	THET	=	θ, Boundary Layer Momentum Thickness
COEFF2	THODEL	=	θ/δ
COEFF2	THOREF	=	θ/Reference Length
EDGPRP	UE	=	U_e/U_{ref}
EDGPRP	UERO2	=	$(U_e/U_{ref})r^{2j}$
EDGPRP	UESO	=	U_e/U_{ref} at s = 0

ARRAY2	UEZ	=	U_e/U_∞, Array
FRSTRM	UFS	=	U_∞, ft/sec
REF	UREF	=	$U_{ref} = U_\infty$, ft/sec
ARRAY1	VC	=	$V = \int_0^\eta (-2\xi \frac{\partial F}{\partial \xi} - F) d\eta$, Transformed Normal Velocity Component
COMWLL	VW	,	Value of VC at Wall
GEOME	X	,	Local Surface Distance, x
TRANS	XBAR	,	Measure of Relative Length of Transition (i.e. $X_{\text{End of Transition}} \simeq$ XBAR*$X_{\text{Beginning of Transition}}$)
GEOME	XI	=	ξ, Transformed Surface Distance
TRANS	XIBAR	=	$\bar{\xi}$, Transition Intermittency Factor Coordinate $\bar{\xi}$ = (X-XZERO)/XLAMDA, where XLAMDA = (XBAR-1)* XZERO/4 and XZERO is the Value of x at the Start of Transition
GEOME	XIOLD	,	Previous Value of XI
GEOME	XI2	=	2*XI
GEOME	XJAY	=	XJFAC
GEOME	XJFAC	,	Indicator, Zero for 2-D Flow, One for Axially Symmetric Flow
NMLCRD	XKETA	,	Parameter Which Controls the Grid Spacing in the η Direction (1.0 Gives an Equally Spaced Grid, a Value Greater than 1.0 Gives a Grid with a Smaller Spacing at the Wall than at the Outer Edge)

VSCSTY	XK1	,	Constant for Van Driest Inner Eddy Viscosity Law
VSCSTY	XK2	,	Constant for Outer Eddy Viscosity Law
ARRAY1	XN	,	Array of η Values
ARRAY1	XN2	,	Array of η Values (Used only when ETAINF is Increased)
GEOME	XOLD	,	Previous Value of X
COEFF2	XOREFL	=	x/Reference Length
ARRAY2	XSTA	=	x, Array of Surface Distance Values used to Specify Geometry
ARRAY1	Y	=	y/ε_{VD}, Stretched Normal Coordinates
ARRAY1	YOVDEL	=	y/δ
ARRAY1	YOVTHT	=	y/θ
ARRAY1	YY	=	y, Normal Coordinate
GEOME	Z	=	z, Local Axial Coordinate
ARRAY2	ZA	=	z, Array of Axial Coordinate Values used to Specify Geometry
GEOME	ZOL	=	z/Body Length
COEFF2	ZOREFL	=	z/Reference Length
COEFF3	SUM1		Incremental Viscous Drag due to Finite Surface Distance
COEFF3	SUMT		Total Drag due to Skin Friction and Pressure
STAG	RHO		Density from Stagnation/Static Pressure
TVCURV	TVC		Flag for TVC Calculations; Yes for TVC, No for 2-D

TVCURV	BODANG	Local Body Angle
TVCURV	RTVC	Local Body Radius
TVCURV	RORW	TVC Body Radius
TVCURV	RORWSQ	RORW
TVCURV	RW	Local Body Radius, ft.
SPTVC	RNOSE	Nose Radius, ft.
WTE	IW	Counter for IWT
WTE	IWT	Increment for Printing Full Profile Data
TE	KTE	Station Number of Trailing Edge
TE	KTSTTE	Flag for L.E. Station Calculations
WAK	XKFAC	Flag for Making Wake Calculations; 0 for No Wake, 1 Include Wake
HESS	IWRTH	Displacement Body-Original Body Weighting Factor
HESS	ICARD1	Hess Code Execution Parameter, Printed on Bodycoor Data
HESS	ICARD2	
HESS	ICARD3	
HESS	ICARD4	
HESS	NN	Number of Body Stations
HESS	IBDN	Body Sequence
HESS	HEDR	Array of Alphanumeric Characters Describing the Executing Case
HESS	CASE	Alphanumeric for Recording Case Number

HESS	KHESS	Flag: 1 if Inviscid C_p Data are Read in from HESS, 0 if no Inviscid C_p Data are Read in
DRAG	CDV1	Old Value of CDV2
DRAG	CDV2	Pressure Drag for Incremental Length dx
DRAG	XCP	C_p at x from Current Iteration
DRAG	XCP1	C_p at x from Previous Iteration
DRAG	CP	C_p Array from Current Iteration
DRAG	CP1	C_p Array from Previous Iteration
DRAG	ROMAX	Maximum Body Radius
INTACT	BETAS	Minimum BETA
STRTUP	IREAD	Flag for Restart Procedure; 1 use Restart, 0 do not use Restart
STRTUP	ITER	Station Number at Which Restart Tape is Written
STRTUP	ITER	Iteration Number
BDCOOR	ZDELST	Axial Displaced Body Coordinate at Original Axial Coordinate Station
BDCOOR	RODELS	Displaced Body Radius at Station k
BDCOOR	ZOLD	Value of Z from Previous Iteration
BDCOOR	ROLD	Value of R from Previous Iteration
BDCOOR	MM	IRST
APLDAT	XLOCAT	Axial Location of the Upstream N-S Region Boundary

APLDAT	DYDX	Body Slope over the Tail Section, Taken as Constant for the Straight Taper Requirement of the APLNS Code
APLDAT	JSTA	Station Number of the Beginning of the N-S Box
APLDAT	NOPRFS	Number of APLNS Code Stations on the Body

Description of Subroutines.

Subroutine MAIN. The principle function of MAIN is to control the flow of logic in the program; however, some calculations are also made in MAIN. In general, the calculations in MAIN are such that they cannot conveniently be performed in the subroutines or they are not sufficiently similar to the calculations made in particular subroutines.

Subroutine READIN is called first. If the input data are from the restart tape, the calculations are made which normally occur at the end of MAIN after subroutine WRITE2 is called, i.e., the calculations that enable the program to set up for the next station are made. The next section of the program is skipped, and the marching integration is resumed. Otherwise, the initial data and profiles are calculated and written by subroutine WRITE1 and the integration is begun with the statement; DO 330 K=KK, KEND. An iteration loop is then entered for the solution of the velocity profile at that station. The boundary conditions at the wall are calculated and the momentum equation is solved. The transformed normal velocity component V is calculated together with the skin friction in the transformed coordinates (CF). The iteration counters are incremented and NIT and ISTOP are checked. If ISTOP is greater than

100, the program is stopped. If NIT equals (or exceeds) NIT3 the step-size is halved, X and DX are recalculated, and control is returned to the beginning of the iteration loop. Otherwise, the next test is the convergence test. If the convergence criterion is not satisfied, control is returned to the beginning of the iteration loop.

When convergence has been obtained, the velocity profile is checked to see if η_∞ needs to be increased. If so, subroutine ADDETA is called and control is returned to the beginning of the iteration loop. If not, subroutines COEF and WRITE2 are called. New values of x and ξ are obtained and the present values of velocity are assigned to the arrays which contain the values from the previous iteration, and the iteration counters NIT and ISTOP are reset. If a solution has been obtained for the final value of x, (or for K = KEND) the program stops, otherwise, the program returns to the beginning of the marching integration loop to obtain a solution at the new value of x.

Description of Variables.

ABST	=	F2(IE) - F2(IE-4). If ABST is greater than or equal to ADTEST, η_∞ is increased
IPRIIM	=	IPR (II-1)
KREADN	;	Indicator for Subroutine READIN, 1 for Input Data, 2 for Initial Data Written on Restart Tape, 3 for Station Data Written on Restart Tape
N	,	Array subscript

The following variables used in MAIN are described in the section Description of Variables in Common: ADTEST, A1B, CCRNI, CF, CF1, CONVERG CRNI, DIF, DN, DUEDS, DX, EPSVD, FC, FCN, FCP, F1, F1N, F2, F2N, F2NN,

F2N1 IE, II, IIMAX, IPR, IPRNT, ISTOP, K, KADETA, KEND, KSTOP, KSTRT, LAMTRB, NIT, NITTOT, NIT3, NOSE, NT3, PNC, REINF, RHOFS, RO, SCF, UE, UFS, UREF, VC VW, X, XI, XIOLD, XI2, XJFAC, XKFAC, XN, XOLD, XSTA, Y, YY.

Subroutine ADDETA. Subroutine ADDETA is used to increase/decrease ETAINF when the value of F at the outermost grid point and the value of F four points inward differ by more/less than the quantity ADTEST. A new array of values is generated with the maximum value increased by 10 percent. Three point interpolation is used to obtain the values of velocity at the new grid points, whereas at the grid points lying between the new and old values of η_∞, f is set equal to unity. First and second derivatives of F are then calculated. The storage arrays for F and their derivatives are updated and V is recalculated using the new values. The arrays for y are not updated in this subroutine but are recalculated at the normal points in the solution procedure.

The subroutine provides the output of the intermediate profiles following the message that η_∞ has been increased. DX, XKETA, ETAINF, and ADTEST are also included in the output.

USAGE:

CALL ADDETA

Description of Variables.

DETA1	$\Delta\eta$ at wall
ETAIN2	New value η_∞
JC	Array subscript
N	Array subscript

The following variables used in subroutine ADDETA are described in the section Description of Variables in Common: AOBP, CHI, DN, DN2, DX, EPSPL, ETAINF, FC, FCN, FCP, F1, F1N, F1NN, F2, F2N, F2NN, IE, IM, ISTOP KL, VC, XKETA, XN, XN2, YOVDEL, YOVTHT, YY.

Subroutine BLUNT1. This subroutine calculates the edge and reference conditions at the stagnation point for a blunt body. The derivative of U_e with respect to s is obtained using the five-point differentiation formula with two points reflected about the stagnation point.

USAGE:

CALL BLUNT1

Description of Variables.

DUM	Dummy variable
DUM1	Dummy variable
DUM2	Dummy variable
DUM3	Dummy variable
J	Array subscript
NERR	Dummy variable
PE	Array of PL values
UE	Array of U values

The following variables used in Subroutine BLUNT1 are described in the section Description of Variables in Common: BETA, DUEDS, PE, PESO PFS, PNC, PP, PREF, PSTAG, PZ, POPRIM, P10, REINF, RO, SCF, UE, UERO2 UFS, UREF, X, XJFAC, XM, XSTA, Z, ZA.

Subroutine BLUNT2. For a blunt body, the edge properties required by subroutine DELTAS to calculate $\Delta\xi$ are provided by subroutine BLUNT2. In the immediate vicinity of the stagnation point, z, r, Pe and

dU_e/ds are calculated using five point interpolation with two points reflected about the stagnation point.

USAGE:

CALL BLUNT2

Description of Variables.

DUM	Dummy variable
DUM1	Dummy variable
DUM3	Dummy variable
J	Array subscript

The following variables used in subroutine BLUNT2 are described in the section Description of Variables in Common: DUEDS, IIAX, PE, PP PZ, P10, RZ, RO, UE, UEZ, X, XSTA, Z, ZA.

Subroutine CHANGE. Subroutine CHANGE adjusts the x stepsize when necessary to obtain a solution at the specified values of x. Subroutine CHANGE is also used when calculations are made with the instantaneous transition model. In this case LAMTRB is reset to two, the stepsize is reduced and a message is printed.

Subroutine CHANGE also increases or decreases the x-stepsize depending on the number of iterations required to converge the previous station.

USAGE:

CALL CHANGE

Description of Variables.

IPRII	= IPR(II)
IPRIIM	= IPR(II-1)

The following variables used in subroutine CHANGE are described

in the section Description of Variables in Common: DX, DXMAX, DXOLD, DX1, EPSVD,, II, IIMAX, IPR, JJ, KSTOP, DTRNSN, LAMTRB, NIT, NIT1, NIT2, X, XOLD, XSTA.

Subroutine COEF. Subroutine COEF calculates the skin-friction coefficients for the boundary-layer flow, the pressure drag, and total drag. It calculates the Reynolds number, displacement thickness, momentum thickness and other dimensionless parameters. At X = 0 the calculations in this subroutine are omitted except for Z/L.

USAGE:

CALL COEF

Description of Variables.

CF2	Term used in computing CFBREX
DSUM	Term used in computing CFBREX
FAC1	Dummy variables
FAC2	Dummy variables
I	Array subscript
N	Array subscript
RESE	$Re_x \varepsilon_{VD}^2$
SQREX	Square root of REX
STRES	CHREY

The following variables used in subroutine COEF are described in the section Description of Variables in Common: A1B, CF, CFBAR, CFBREX, CFINF, CFRES, CFREY, CF1, DEL, DELORF, DELOX, DELST, DELTA, DN, DSAXOR, DSTODL, DSTORF, DSTOTH, DSTOX, STRAX, DX, EPSVD, FC, HAFCF, IE, IIMAX K, KFS, LAMTRB, REFLEN, REINF, RETHET, REX, PNC, RHOFS, ROREFL, RO SUM, THET, THODEL, THOREF, UE, UFS, X, XJFAC, XOREFL, Y, YOVDEL, YOVTHT, YY,

Z, ZA, ZOL, ZOREFL.

Subroutine DELTAS. Subroutine DELTAS calculates the transformed surface distance ξ using integration by Simpson's rule. A value of ξ is obtained for each increment $\Delta\xi$ from the equation

$$d\xi = U_e r_w^{2j} U_e^{2k} dx$$

The transformation factor PNC,

$$PNC = \frac{\sqrt{2\xi}}{U_e r^j}$$

and the pressure gradient term, β,

$$\beta = \frac{2\xi}{U_e} \frac{dU_e}{d\xi}$$

are also calculated.

USAGE:

CALL DELTAS

Description of Variables.

HA	X - DX/2
RP	RO at X
ROHA	UE at HA
X1	Storage value of X

The following variables used in subroutine DELTAS are described in the section Description of Variables in Common: BETA, DS, DUEDS, DX, PNC, RO, UE, UERO2, X, XI, XIOLD, X12, XJFAC, XKFAC.

Subroutine DERIV3. Subroutine DERIV3 uses subroutine FD3 to generate the first derivative of a function which is given by the

array F. The array of abscissa values is given by X and the derivative of F with respect to X is returned in the array FP.

USAGE:

CALL DERIV3 (F,X,IMAX,IMIN,FP) where F and X are the arrays of ordinate and abscissa values, and where IMAX and IMIN are the upper and lower subscripts for the array FP.

Subroutine EFFMU. Subroutine EFFMU calculates the terms in the coefficients of the differential equations which contain the eddy viscosity. These terms are A_0 and $\frac{\partial A_0}{\partial \eta}$ where:

$$A_0 = (1+\varepsilon^+)$$

for the momentum equation. For the laminar case, $\varepsilon^+ = 0$ and the terms reduce to

$$A_0 = 1$$

and

$$A_0' = 0$$

The first portion of the subroutine calculates the terms for the laminar case, and the next portion of the subroutine calculates the vorticity Reynolds number, x,

$$x = \frac{y^2}{\nu} \frac{\partial u}{\partial y}$$

For calculations with a transition regime, the routine initiates transition and calculates the transition intermittency factor Γ where

$$\Gamma = 1 - \exp(-0.412\overline{\xi}^2),$$

$$\overline{\xi} = 4 \frac{x-x_0}{(\bar{x}-1)x_0}$$

and x_0 is the location of the beginning of transition and $\bar{x}$ is a measure of the relative length of transition.

In the third portion of the subroutine, the eddy viscosity is computed for turbulent flow calculations and the terms A_0 and A_0' are then computed.

USAGE:

CALL EFFMU (LAMTRB, II, DSTARK)

Description of Variables.

ARG	Dummy variable
DELTA	δ/ε_{VD}, boundary-layer thickness in stretched coordinates
DSTARK	δ_k^*, incompressible displacement thickness
FAC	Dummy variable
FAC1	Dummy variable
FAC2	Dummy variable
FAC3	Dummy variable
FAC4	Dummy variable
GAMMA	γ, outer eddy viscosity intermittency factor
I	The subscript in the EPSPL array where the inner and outer eddy viscosities are matched.
LAMTRB	Indicator for laminar or turbulent flow; 1 for laminar, 2 for turbulent
N	Array subscript
XLAMDA	$\lambda = (\bar{x}-1)x_0/4$
XZERO	X_0, location of start of transition

The following variables used in subroutine EFFMU are described in the section Description of Variables in Common: ATR, AOB, AOBP, CF,

CHI, CHICRT, CHIMAX, DN, DX, DXOLD, EPSO, EPSPL, EPSVD, FC, FCN, GAMMA, PRT, RO, UE, X, XBAR, XIBAR, XI2, XJFAC, XKFAC, XK1, XK2, XN, Y.

Subroutine EDGPROP. Subroutine EDGPROP obtains the edge properties used in subroutine DELTAS to calculate $\Delta\xi$. The edge properties are obtained from subroutine BLUNT1.

USAGE:

CALL EDGPROP

Subroutine FD3 and FD5. Subroutines FD3 and FD5 use the first derivative of the Lagrangian interpolating polynomial of second and fourth order, respectively, to provide the first derivative of the function F at point X. The returned value is denoted by FX. The interpolating polynomials are described in the descriptions of subroutines INTER3 and INTER5.

USAGE:

CALL FD3 (X,X1,X2,X3,F1,F2,F3,FX) or

CALL FD5 (X,X1,X2,X3,X4,X5,F1,F2,F3,F4,F5,FX) where X1, X2, etc. are the abscissa values, and where F1, F2, etc. are the ordinate values.

Subroutine GEOM. This subroutine obtains the edge properties at the initial value of x for the geometry specified by the variable NOSE which is passed through common. The edge properties are obtained from subroutine BLUNT1 for a blunt nosed body (NOSE = 1), from subroutine CONE1 for a sharp nose (NOSE = 2).

USAGE:

CALL GEOM

Subroutine INIT. Many variables are given fixed initial values in the program before the marching integration procedure is begun.

The non-subscripted variables are given the initial values in subroutine INIT.

USAGE:

CALL INIT

Description of Variables.

JT Array subscript

The following variables used in subroutine INIT are described in the section Description of Variables in Common: CCRNI, CF, CHIMAX, CRNI, DS, DX, DXOLD, DX1, E1, FF, GAMMA, IE, IIMAX, IM, IPRNT, ISTOP, JJ, KEP, KFS, KSTOP, KTPW, KTW, LAMTRB, NIT, NITTOT, NOSE, NRSTRT, RHOFS, SUM, UFS, VW, X, XI, XIBAR, XIOLD, XI2, XJAY, XJFAC, XKFAC, XM, XOLD, XSTA, Z.

Subroutine INTER3. Subroutine INTER3 uses a second-order Lagrangian interpolating polynomial, interpolating on the points x1, x2, and x3, with the corresponding function values F1, F2, F3 to provide a value F(x).

The general form of the polynomial is

$$F(x) = \sum_{k=1}^{3} f_x L_k(x)$$

where

$$L_k(x) = \prod_{\substack{m=1 \\ m \neq k}}^{3} \frac{x - x_m}{x_k - x_m}$$

USAGE:

CALL INTER3 (X, X1,X2,X3,F1,F2,F3,F) where X1, X2, and X3 are the abscissa values and F1, F2, and F3 are the function values.

Subroutine INTER5. Subroutine INTER5 uses a fourth-order Lagrangian interpolating polynomial interpolating on the points x1, x2, x3, x4, and x5 with the corresponding function values F1, F2, F3, F4, and F5 to provide a value F(x).

The general form of the polynomial is

$$F(x) = \sum_{k=1}^{5} f_x L_k(x)$$

where

$$L_k(x) = \prod_{\substack{m=1 \\ m \neq k}}^{5} \frac{x - x_m}{x_k - x_m}$$

USAGE:

CALL INTER5 (X,X1,X2,X3,X4,X5,F1,F2,F3,F4,F5,F) where X1, X2, etc. are the abscissa values and F1, F2, etc. are the function values.

Subroutine INTERP. Subroutine INTERP uses function TLU to linearly interpolate in the array F2 for the value FF corresponding to the value XX in the array XN. If XX is not within the range of XN, a message is printed and FF is set equal to F2(IE) where IE is the size of the arrays. XN must be a strictly increasing array.

USAGE:

CALL INTERP (XX,XN,F2,IE,FF)

Subroutine INTERP5. Subroutine INTRP5 uses five-point interpolation to obtain a value F0 in the array F1 corresponding to the value X0 in the array X1. The array X1 can be either monotone increasing or monotone decreasing.

USAGE:

CALL INTRP5 (XO,X1,F1,IMAX,IMIN,I1,FO) where X1 and F1 are the arrays of coordinate values, IMAX and IMIN are the maximum and minimum values allowed for the subscripts of the arrays O and I1 is an indicator ($I1 \leq 0$, X1 is monotone decreasing; $I1 \geq 0$, X1 is monotone increasing).

Subroutine MOMENT. The solution of the momentum equation is obtained in subroutine MOMENT. The coefficients of the partial differential equation are calculated and new values of $F = U/U_e$ and the first and second derivatives of F with respect to η are obtained from subroutine SOLVE. The difference between the new and the former values is obtained for either the wall or the all points convergence test. The derivative of F with respect to x is also calculated.

USAGE:

CALL MOMENT

The following variables in subroutine MOMENT are described in the section Description of Variables in Common: AO, AOB, AOBP, A1, A2, A3, A4, BETA, C, CCRNI, CP, CRNI, DIFF, DS, DSTARK, E1, FC, FCN, FCP, FF, F1, F1N, F1NN, F2, F2NN, F2N1, IE, IM, KEP, LAMTRB, NC, TH, VC, X, XI.

Subroutine PRFILE. Subroutine PRFILE calculates the array of η values, XN, which corresponds to the values of IE, ETAINF, and XKETA. The grid spacing is given by $\Delta\eta_i = K\Delta\eta_{i-1}$ where K corresponds to XKETA and thus at the ith grid point

$$\eta_i = \Delta\eta_1 \frac{K^i-1}{K-1}, \quad i = 0, 1, 2, 3, \ldots N \ (N = IE - 1)$$

where $\Delta\eta_1$ is given by

$$\Delta\eta_1 = \eta_\infty \frac{K-1}{K^N-1}$$

The initial velocity profile, F, is calculated from the formula $F = 1 - \exp(-\eta)$. The derivatives of F are also calculated as are initial profiles for ε^+, y/δ, y/θ. Further, the boundary conditions at the outer edge of the boundary layer are set in this subroutine.

USAGE:

CALL PRFILE

Description of Variables.

DETA1 η stepsize at wall

N Array subscript

The following variables used in subroutine PRFILE are described in the section Description of Variables in Common: DN, EPSPL, ETAINF, FC, FCN, FCP, F1, F1N, F1NN, F2, F2NN, IE, IM, PNC, VW, XKETA, XN, Y, YOVDEL, YOVTHT.

Subroutine READIN. Subroutine READIN provides the input of data for the program. Input data are from a single data set (SUB DATA) when calculating a noniterative solution, two sets for the initial iterated solution (SUB DATA and HESSCP OUTPUT), and the two plus a restart tape for the subsequent iterations. READIN also resets the input flags to their default value when they are set to 0 in the input data and performs some pressure calculations for the iterated solution.

USAGE:

CALL READIN

Description of Variables.

J Array subscript

LST Alphanumeric variable, assigned value of 'LAST' in a DATA statement

All other variables are described in the section Description of Variables in Common, except for LABEL and LSTC which are described in the section.

Description of Input Data.

Subroutine RESTAR. Subroutine RESTAR restarts program execution at a predetermined body coordinate location. On the first iteration, when the marching integration reaches the restart location (IRSTRT), a tape is written containing all the data needed to begin execution on the next global iteration. RESTAR first reads in the data written from the previous global iteration, and then reads in the previous displacement body coordinates. Finally the restart tape is rewritten using the data calculated by the current execution and the coordinates of the new displacement body are written out at the end of the current execution.

Subroutine SOLVE. Subroutine SOLVE calculates the solution for a general parabolic partial differential equation when the equation is written in the standard form.

$$\phi_{\eta\eta} + A_1\phi_\eta + A_2\phi + A_3 + A_4\phi_\xi = 0$$

In order to obtain a solution for the above equation, values of ϕ and the first and second derivatives of ϕ with respect to η from a previous iteration or from a previous x station are required. These data are passed through the argument list as the arrays W1, W1N, and W1NN. The new values of ϕ and the first and second derivatives are returned through the argument list as the arrays W2, W2N, and W2NN.

USAGE:

CALL SOLVE (W1NN, W1N, W2NN, W2N, W2, E1, F11, CRNI)

Description of Variables.

A	Matrix Inversion Coefficient
B	Matrix Inversion Coefficient
CC	Matrix Inversion Coefficient
D	Matrix Inversion Coefficient
E	Array of Matrix Inversion Coefficients
E1	Value of ε at wall
F	Array of Matrix inversion coefficients
F11	Value of F at wall
KON	Array subscript, descending order
N	Array subscript
W1	Value of ϕ at previous station
W1N	$\partial\phi/\partial\eta$ at previous station
W1NN	$\partial^2\phi/\partial\eta^2$ at previous station
W2	New value of ϕ
W2N	New value of $\partial\phi/\partial\eta$
W2NN	New value of $\partial^2\phi/\partial\eta^2$

The following variables which are used in subroutine SOLVE are described in the section Description of Variables in Common: A1, A2, A3, A4, CRNI, DN, DS, IE, IM, XN.

Function TLU. Function TLU searches in the array X for the two values which bracket XSTAR and linearly interpolates for the corresponding value in the array Z. The returned value is TLU. If XSTAR is not within the range of the array X, TLU is set equal to zero and the error flag is set equal to one. The array X must be strictly increasing.

USAGE:

TLU (NTABLE, Z, X, XSTAR, NFLAG) where NTABLE is the dimension of the arrays Z and X and NFLAG is the error flag.

Subroutine VELDAT. Subroutine VELDAT has been added for the express purpose of writing out data sets for the Navier-Stokes code. The routine writes the solution of the velocity profile at the upstream boundary of the Navier-Stokes region to be used as boundary conditions for the APLNS code.

USAGE:

CALL VELDAT

Subroutine WRITE1. Subroutine WRITE1 writes out the input data and the initialized data. Integer quantities are written first in the same order as they appear in the input. The next data written are the freestream quantities. The transition parameters are then written, followed by a group of variables, most of which are from the program input and appear in approximately the same order. Next are written the integer arrays which control the values of x at which a solution is obtained and stations at which full profile data are to be written. Finally, the values of axial distance, radius, surface distance, surface pressure, and nondimensional edge velocity are written.

USAGE:

CALL WRITE1

Description of Variables.

J Array subscript

The other variables which appear in this subroutine are described in the sections Description of Output and Description of Variables in Common.

Subroutine WRITE2. Subroutine WRITE2 provides the output of computed results after a converged solution is obtained. The output is in two sections. The first section is written at every station at which a solution is obtained. The second section is written for specified values of x. The first line of output contains the values of geometry related variables (such as x, z, and ξ), the pressure gradient term β, the number of iterations, and the station counter, K. The second line contains the nondimensionalized edge conditions (U_e, P_e, etc.), the skin friction in transformed coordinates, the matched eddy viscosity, and the cumulative total of the number of iterations. The third line gives the first and second derivatives of the velocity at the wall and at the second grid point from the wall. The remainder of the first section is omitted at x=0. The next line gives transition parameters and the following line contains the Reynolds number. In the remaining lines in the first section are skin friction coefficients, the boundary-layer thicknesses, and quotients of boundary-layer parameters. In the second section of the output, some of the variables which are functions of the normal coordinate are written. The output of this section contains η, y, y/θ, y/δ, $F(=U/U_e)$, $\partial F/\partial\eta$, y^+, u^+, Udef, and ε^+.

USAGE:

CALL WRITE2

Description of Variables.

II	Counter or subscript for array IPR
IPRNT	Counter or subscript for array IPRFL
KL	The values of η and the functions of η are written

at every KLth grid point

XOLD Previous value of x

The above variables are in common and are also described in the section Description of Variables in Common. The other variables which appear in this subroutine (except for N, an array subscript) are described in the section Description of Output.

Subroutine ZRO. Subroutine ZRO uses subroutine INTER5 to interpolate in the arrays ZA and RZ for the values Z and RO corresponding to the value X2 in the array XSTA. The arrays RZ, XSTA, and ZA contain values of radius, surface distance, and axial distance respectively for an axisymmetric geometry.

USAGE:

CALL ZRO (JJ, X2, Z, RO)

Description of Input Data.

There are two sets of input files to be used in obtaining the iterated solution, i.e., SUB DATA, and HESSCP OUTPUT. SUB DATA contains the basic data for execution, and HESSCP OUTPUT contains the C_p distribution at the edge of the displaced body. SUB DATA will be described first.

CARD 1 IREAD, IRSRT, ITER, KHESS (6(8X,I2))

IREAD	Restart flag, 0 no restart tape to be read (or written), 1 read (and write) restart tape
IRSRT	Station number for restart tape to be written
ITER	Iteration counter

	KHESS	Flag for using C_p distribution; 0 do not read in C_p data from HESS, 1 read in C_p data
CARDS 2-6	HEDR, CASE, ICARD2, ICARD3, ICARD4, NN, IBDN (10A6, 2x, A6/I3/I5/1x,I4,I5/9x,I1) Standard HESS input parameters, described in HESS PARMDATA and HESSCOOR DATA	
CARD 7	BETAS, ROMAX	(6F12.6)
	BETAS	Minimum value of BETA, the pressure gradient term. If BETA goes lower than BETAS, BETA is set equal to BETAS.
	ROMAX	Maximum body radius
CARD 8	LABEL	(18A4)
	Alphanumeric data for identification of the case. Printed on the first page of output.	
CARD 9	NOSE, LAMTRB, KVSAW, KTRANS, KTRNSN, KONSET	(3(8x,I2), 3(7x,I3))
	NOSE	Indicator for geometry; 1 for a blunt nose, 2 for a sharp nose body. A value of 0 is reset to 1
	LAMTRB	Indicator; 1 for laminar flow, 2 for turbulent flow. A value of 0 is reset to 1

	KVSLAW	Indicator: 0 for Reichardt Eddy Viscosity, 1 for Van Driest Eddy Viscosity
	KTRANS	Indicator for transition regime; 0 for no transition or instantaneous transition, 1 for transition regime (LAMTRB must be 1)
	KTRNSN	Subscript for XSTA array giving value of x at which instantaneous transition occurs
	KONSET	Subscript for XSTA array giving value of x at which transition regime begins
CARD 10	NIT1, NIT2, NIT3, NC	(4(8X,I2))
	NIT1	After solution converges, Δx is doubled if NIT is less than or equal to 0. A value of 0 is reset equal to 3. A value less than zero is reset equal to 0. Recommended value is 5
	NIT2	After solution converges, Δx is halved if NIT is greater than or equal to NIT2. A value of 0 is reset to 6. A recommended value is 10
	NIT3	If NIT = NIT3, Δx is halved, X and XI are recomputed and the iteration loop is restarted with the smaller values of X and XI. A value of 0 is reset to 9. A recommended value is 20
	NC	Specific convergence test; 0 for test on derivatives at wall, 1 for test on functions at all points in profile. Recommended value is 1
CARD 11	IE, KEND, IIMAX	(4(7X,I3))
	IE	Number of grid points in ETA direction. (program dimensioned for 101, the recommended value)

	KEND	Upper index on K (the station counter). Recommended value, 300-600
	IIMAX	Number of values in geometry and pressure arrays. Arrays dimensioned for 500 values, recommend no more than 101. If more than 101 points are used, changes will need to be made to the dimensions of arrays IPR and IPRFL
CARD 12	KADETA, KL, IPFL, IPRINT, IWT	(5(8X,I2))
	KADETA	Indicator; if zero, ETAINF is held constant, if nonzero, ETAINF can be increased or decreased. A nonzero value is recommended
	KL	Index for profile output DO loop; 1 gives output for each grid point, 2 for every second grid point
	IPFL	Number of values in IPRFL array (maximum 100)
	IWT	Increment in station number at which full profiles are printed
CARD 6	CHICRT, XBAR, ATR	(3F12.6)
	CHICRT	Value of CHIMAX at which transition regime calculations begin (arbitrary if KTRANS = 0). Appropriate values have been found to be between 2000 and 4000
	XBAR	Measure of relative length of transition region (i.e. $X_{\text{end of transition}} = XBAR*X_{\text{beginning of transition}}$) For Reichardt Law, XBAR should be about 65 percent greater than value used with Van Driest Law. Recommended values are approximately 3.3 and 2.0, respectively

	ATR	Transition constant = 0.412
CARD 14	UFS, REINF, PFS, PSTAG, SCF, RNOSE	(6F12.6)
	UFS	Freestream velocity, ft/sec
	REINF	Freestream Reynolds number per unit length
	PFS	Freestream static pressure, psfa
	PSTAG	Freestream stagnation pressure, psfa
	SCF	Scale factor to convert unit length for REINF (1.0 if REINF/ft, 12.0 if REINF/in. etc.)
	RNOSE	Nose radius, ft
CARD 15	XK1, XK2, CONVRG, ADTEST	(6F12.6)
	XK1	Constant for Van Driest Inner Eddy Viscosity Law. A value of 0.0 is reset to 0.4, the common value
	X	Constant for Outer Eddy Viscosity Law. A value of 0.0 is reset to 0.0168, the common value
	CONVRG	Convergence criterion. A value of 0.0 is reset to 0.001, the recommended value
	ADTEST	Criterion for increasing ETAINF (F(IE))-F(IE-4) is compared with ADTEST.) A value of 0.0 is reset to 0.001, the recommended value
CARD 16	DX, CRNI, XKETZ, ETAINF, REFLEN	(6F12.6)
	DX	Initial x stepsize
	CRNI	Selector for finite difference scheme; 1.0 for fully implicit

		(recommended), 0.5 for Crank-Nicolson, and 0.0 for fully explicit (not recommended)
	XKETA	Parameter which controls the grid spacing in the ETA direction (1.0 gives an equally spaced grid, a value greater than 1.0 gives a finer grid at the wall than at the outer edge.) A value of 0.0 is reset to 1.09, the value recommended for IE = 101
	ETAINF	Maximum value of ETA. A value of 0.0 is reset to 100 which is the recommended value for a flow which is initially turbulent. For an initially laminar flow this value should be set to 6.0)
	REFLEN	An arbitrary reference length
CARD 17	XJFAC, XKFAC, TVC	(2F12.6,4X,A3)
	XJFAC	Indicator; 0.0 for 2-D flow, 1.0 for axially symmetric flow
	XKFAC	Indicator; 0.0 for body only solution, 1.0 for body and wake solution
	TVC	Indicator; YES to include TVC effects, NO without TVC
CARD 18	DXMAX, XLOCA, DYDX, DX, NOPRFS	
	DXMAX	Maximum x stepsize permitted in the calculations
	XLOCAT	Location on the body of the upstream edge of the Navier-Stokes region
	DYDX	Slope of straight taper fit of tail of body
	DX	x stepsize for use in the Navier-Stokes code
	NOPRFS	Number of profiles on the body in the Navier-Stokes region

CARDS 19a, b, c...	(IPR(J),J = 1, IPRINT) (14I5)
IPR	Array of subscript for array XSTA giving values of x at which complete solutions (with profiles) are printed (array dimensioned for 100 values)
CARDS 20a, b, c...	(IPRFL(J), J = 1, IPFL) (14I5)
IPRFL	Array of subscripts for array XSTA setting values of x at which complete solutions (with profiles) are printed (array dimensioned for 100 values). Also at these stations are printed the coordinates of the displaced body. Each value of IPRFL must also be a value of IPR
CARDS 21-21+IIMAX	ZA(J), XSTA(J), RZ(Z), PZ(J), LSTC (4F12.6,A4)
ZA	Array of coordinate values along the body axis
XSTA	Array of coordinate values along the body surface (same as ZA for a flat plate)
RZ	Array of radius values for a body
PZ	Array of pressure ratio values - P/P_0 where P_0 is the local stagnation pressure. (These are always 0.0 for the iterated solution and are given values only when KHESS = 0)
LSTC	Alphanumeric variable used to indicate the last data card. LAST is coded on the final card.

See Appendix B, Table 2.1 for a sample SUB DATA data set.

Description of HESSCP OUTPUT.

CARD1-IIMAX	X(J), CP(J), J = 1, IIMAX (2F12.6)

X	Axial coordinate of body point, ft
CP	Pressure coefficient at body point

See Appendix B, Table 2.2 for a sample HESSCP OUTPUT data set.

Description of Output Data.

ICBLINT outputs three data sets. The first is the complete boundary-layer solution over the body (BLAYER OUTPUT). The second is the velocity profile at the upstream edge of the Navier-Stokes region with information on the profile included (APLNS VELDATA). The third set is the coordinates of the displaced body (BODYCOOR DATA).

This section contains a description of BLAYER OUTPUT in three subsections. The first subsection describes the first section of the output which is the input and initialized data. The second subsection describes the data which are written for each station at which solutions are obtained. The third subsection describes the miscellaneous messages and data written by the program. The output from a sample calculation of turbulent flow over a cylinder is included at the end of this section (see Appendix C, Table 1.1).

Input and Initialized Data.

LABEL		,	Alphanumeric Data for identification of the case (written by Subroutine READIN)

The data in the remainder of the first block is written by subroutine WRITE1 and corresponds to the input data with the following exceptions:

Line 6	RHOFS	=	ρ_∞, slugs/ft^3
Line 10	EPSVD	=	$\epsilon_{VD} = \{\mu_{ref}/(\rho_\infty U_\infty L)\}^{\frac{1}{2}}$, where L is the unit length in REINF

Next in the program output are the arrays IPR and IPRFL (which are described in the section "Description of Input"). Following the arrays IPR and IPRFL are the arrays ZA, RZ, XSTA, PZ, and UEZ with the array subscripts. PZ in the output differs from the values of PZ in the input in that for input PZ is P/P_0 and is changed by the program to P/P_{ref}. UEZ is the array of nondimensional edge velocities calculated by the program (U_e/U_∞). For a flat plate, the values of UEZ are unity.

Station Data.

The station data are divided into two groups. The first group of data is written for each value of x at which a solution is obtained. The second group --the function profiles--is written only for the values of x corresponding to XSTA (IPFRL). In the first group, lines 5-13 are not written if x = 0. In the sixth line, REX alone is written if KFS = 0. If KFS ≠ 0, REX is written followed by the wall heat transfer (BTU/ft^2/sec) and two heat transfer coefficients; one in BTU/in^2-sec-°R, the other in lbm/in^2-sec.

GROUP ONE

Line 1	S	,	Surface Distance, x
	XI	=	ξ, Transformed Surface Distance
	Z	,	Axial Distance, z
	RO	,	Local Radius, r, (Set Equal to One for a Wedge or a Flat Plate)
	BETA	=	$\beta = 2\xi/U_e(dU_e/d\xi)$
	PP	=	dPE/ds
	NIT	,	Number of Iterations at Current Value of x
	K	,	Station Counter

Line 2	UE	=	U_e/U_∞
	PE	=	P_e/P_{ref}
	DUEDS	=	dUE/ds
Line 3	CF	,	Skin Friction in Transformed Coordinates = $C_{f_\infty}/\varepsilon_{VD}$
	EPSVD	=	$\varepsilon_{VD} = \{\mu_{ref}/(\rho_\infty U_\infty L)\}^{1/2}$, Where L is the Unit Length in REINF
	KEP	,	Subscript in EPSPL Array Where the Inner and Outer Eddy Viscosities are Matched (Zero for Laminar Flow)
	NITTOT	,	Cumulative Total Number of Iterations in x Direction
Line 4	(F = U/U_e; Prime Denotes $\partial/\partial\eta$)		
	F2N(1)	=	F'_w
	F2N(3)	=	F' at Two Points from the Wall
	F2NN(1)	=	F''_w
	F2NN(3)	=	F" at Two Points from the Wall
	UPLUS	=	Friction Velocity
Line 5	CHIMAX	,	Maximum Value of Vorticity Reynolds Number - χ
	GAMMA	,	Transition Intermittency Factor (Set to Zero for Laminar Flow, Set to One for Turbulent Flow)
	XIBAR	=	$\overline{\xi}$, Transition Intermittency Factor Coordinate; Initialized as Zero, Calculated after Transition Regime is Entered
Line 6	REX	,	Local Reynolds Number Based on Edge Conditions

WALL HEAT TRANSFER AND HEAT TRANSFER COEFFICIENTS (If KFS≠0)

Line 7	CFE	=	$C_{f_e} = 2\tau_w/\rho_e U_e^2$
	CFINF	=	$C_{f_\infty} = 2\tau_w/\rho_\infty U_\infty^2$
	CF(TOTAL)	=	$\overline{C}_f = \frac{1}{x}\int_0^x C_{f_\infty}\, dt$
	CFBAR*SQRT(REX)	=	$\overline{C}_f \sqrt{REX}$
	CFINF*SQRT(REX)	=	$C_{f_\infty} \sqrt{REX}$
Line 8	CFINF/2	=	$C_{f_\infty}/2$
Line 9	X/REFLEN	=	x/reference length
	Z/REFLEN	=	z/reference length
	RO/REFLEN	=	r/reference length
	DELTA/X	=	δ/x
	DELSTR/X	=	δ^*/x
Line 10	THETA/REFLEN	=	θ/reference length
	DELSTR/REFLEN	=	δ^*/reference length
	DELTA/REFLEN	=	δ/reference length
	DELSTRAXI/ REFLEN	=	δ^*_{axi}/reference length
Line 11	THETA/DELTA	=	θ/δ
	DELSTR/DELTA	=	δ^*/δ
	DELSTR/THETA	=	δ^*/θ
	DELSTRAXI	=	$\delta^*_{axi} = r\,\{\sqrt{1+2\delta^*/r} - 1\}$ (Zero for 2-D Flow)
Line 12	DELSTRK	=	δ^*_k, Incompressible Boundary-Layer Displacement Thickness Computed Only for Turbulent Flow

	DELSTR	=	δ^*, Boundary-Layer Displacement Thickness
	DELTA	=	δ, Boundary-Layer Thickness (Value of y where $U/U_e = 0.995$)
	THETA	=	θ, Boundary-Layer Momentum Thickness
	RETHETA	=	Re_θ, Edge Unit Reynolds Number Times θ
GROUP TWO		PROFILES	
Block 1	ETA	,	η, Transformed Normal Coordinate
	Y	,	y, Corresponding Physical Coordinate
	Y/THETA	,	y/θ (Zero for x = 0)
	Y/DELTA	,	y/δ (Zero for x = 0)
	F=U/UE	,	Non-dimensional Velocity
	FP(N)	=	$\frac{\partial F}{\partial \eta}$
	YPLUS	=	Law of the Wall Normal Coordinate, yU^+/ν
	U/UPLUS	=	U/U^+ Where $U^+ = (\tau_w/\rho)^{\frac{1}{2}}$
	UDEF	=	Defect Velocity, $\frac{U-U_\infty}{U^+}$
	EPS+	=	Eddy Viscosity (Zero for Laminar Flow)
	PITOT	=	The local pitot pressure in p.s.t.
	N	,	Grid Point Number

Miscellaneous Messages and Data.

Subroutine READIN writes the message "RESTART TAPE WRITTEN. K = # KTPW = #" after the station data has been written on the restart

tape. K is the corrent value of K and KTPW is the next value of I at which the restart tape will be written.

For transition regime calculations, subroutine EFFMU writes the message "TRANSITION BEGINS" at the beginning of transition and the message "TRANSITION ENDS" when XIBAR first exceeds 2.0 with the Reichardt Eddy Viscosity Law or when XIBAR first exceeds 4.0 with the Van Driest Eddy Viscosity Law. The message "TRANSITION ENDS" was found to be useful when the Van Driest Eddy Viscosity Law was used in calculations but less useful with the Reichardt Eddy Viscosity Law. At best, the message indicates only approximately where transition ends.

For the instantaneous transition model, subroutine CHANGE writes the message "TRANSITION BEGINS INSTANTANEOUSLY" and the values of X and DX when transition is initiated.

If the iteration counter NIT reaches the value of NIT3, the stepsize is halved, new values of X and XI and new edge properties are calculated and the program attempts to obtain a converged solution for the smaller value of X. In this case, MAIN writes the values of NIT, DX, X, F2N(1), F2N1, and DIF.

When ETAINF is increased, subroutine ADDETA writes the message "INTERMEDIATE PROFILE DATA--ETAINF INCREASED ISTOP = #", and the values of DX, XKETZ, ETAINF, and ADTEST.

If ISTOP exceeds 100, the message "STOP *** ISTOP.GT.100" is written by MAIN and the program is stopped. If after a converged solution is obtained, the derivative of F at the wall is negative, subroutine WRITE2 writes the message "PROBLEM TERMINATED. NEGATIVE DF/DETA INDICATES THAT THE BOUNDARY LAYER HAS SEPARATED" and stops the program.

Normal termination of the program is indicated by the message "THE END X = XSTA(IIMAX)" or the message "THE END K = KEND" which is written twice by MAIN.

In the above descriptions, all quantities which have the apparent dimension of length have been nondimensionalized by the per unit length in REINF.

APLNS VELDATA contains the needed parts of the boundary-layer solution at the beginning of the Navier-Stokes region. All information in it is found in BLAYER OUTPUT and is written again for convenience in using the APLNS program.

BODY COORDATA contains the coordinates of the displaced body calculated during the current global iteration. This data set is used by program HESS for calculating a new pressure distribution around the body.

See Appendix B, Table 2.3 for a sample BODY COORDATA data set.

DESCRIPTION OF PROCEDURE DEVELOP

After the viscous/inviscid solution has been completed, the final displaced body coordinates are used in DEVELOP EXEC to generate the inviscid, displaced flowfield. HESS is run to obtain the flow quantities at off body points (u & v components) for use in the APLNS program input conditions. Since the APLNS code requires 1800 points (as currently dimensioned) and HESS is capable of solving only 400 points per execution, several executions of HESS are required. Each execution supplies a strip of data for input to APLNS. The off-body point locations are developed and the data set written out in DATFIELD using the displaced coordinates from BODYCOOR DATA. DEVELOP EXEC controls the execution of FIELD and HESSBLI EXECs. Execution is in a loop, nominally the loop is executed five (5) times, DATFIELD is executed first, then with the data set written, HESSBLI is run, writing STREAM DATA. Subsequent iterations add data to STREAM DATA until the required full flowfield is completed (1800 points).

See Appendix A, Table 4.1 for a listing of DEVELOP.

Program DATFIELD

Description of Program DATFIELD. FIELD EXEC controls program DATFIELD. DATFIELD writes out the data sets for the execution of HESS to obtain off-body flow conditions. Input data include the final displaced body from HESS/ICBLINT, and the geometry of the afterbody from DATAMA. DATFIELD then calculates and writes the off-body points, together with the displaced body coordinates on HESSCOOR DATA, the HESS data set.

See Appendix A, Table 4.2 for a listing of DATFIELD FORTRAN.

(See Description of Output for a list of variables used in the program.)

Description of Input and Output.

Input Data. DATFIELD uses two input sets: INTER DATA and BODYCOOR DATA. INTER DATA has been described in Output for DATAMA. BODYCOOR DATA has been described in Output of ICBLINT.

Output Data. DATFIELD's output is in two sets: INTER DATA and HESS COORDATA. INTER DATA is rewritten to show the iteration count for DEVELOP. Only the counter, IHESSI is changed. HESS COORDATA contains the HESS data set with the off-body points where a solution is desired.

Card 1	HEDR, CASE	(15A4,2x,A4)
	HEDR	Alphanumeric array describing the body that is running
	CASE	Alphanumeric array defining the case run
Card 2	NB,NNU,NAXI,NCF, NOFF	(5I1)
	NB	Number of bodies input
	NNU	Number of nonuniform flows
	NAXI	Flag for axisymmetric flows; 0 if not axisymmetric, 1 if axisymmetric
	NCF	Flag for crossflow case; 0 if no cross flows included, 1 if crossflows included
	NOFF	Flag; 0 for no off-body points, 1 for off-body points
Card 3 blank		
Card 4	IGEOMF, ISIGF, ICURVN, NONEWF, IFORMT, NN1	
	IGEOMF	Flag for curved elements; 0 for no curvature, 1 for curvature

	ISIGF	Flag for source densities; set to 1; piecewise-linear
	ICURVN	Flag for computation of element curvatures; 0 for input curvatures, 1 for internal computation
	NONEWF	Number of new crossflows for the same body
	IFORMT	Flag; if 0, read first the x, then the y coordinates; if 1, read x,y coordinates in 2F12.6 format, if 2 read x,y coordinates in F12.6,12x,F12.6 format
	NN1	Number of off-body points
Card 5	IBDN	
	IBDN	Body sequence number
Cards 6-NN	X,Y	
	X,Y	Body coordinates
Card NN+7	IGEOMF, ISIGF, ICURVN, NONEWF, IFORMT, NN1	
	IGEOMF	Flag for curved elements; 0 for no curvature; 1 for curvature
	ISIGF	Described above
	ICURVN	Described above
	NONEWF	Described above
	IFORMT	Described above
	NN1	Described above
Card NN+8	IIBDN	
	IIBDN	Body sequence number; 0 if off-body points follow
Card NN+8-	XB, YB	
	XB, YB	, Coordinates of off-body points

See Appendix B, Table 3.1 for a sample HESS COORDATA data set.

Program APLNS

Description of Program APLNS. Program APLNS solves the complete Navier-Stokes equations using an Alternating Difference Implicit (ADI) scheme. Surrounding boundary conditions are required to initiate execution. The boundary conditions at the upstream boundary are held constant while the remainder of the flowfield is iterated upon. The iteration scheme is discussed in more detail later. Due to the large grid size used within the program, there are very few points within the boundary-layer region. Therefore, the results, particularly on the body, must be viewed with caution. To increase the accuracy of the results obtained from the APLNS program, the user is advised that the size of the region of interest be enlarged, i.e., the array sizes be increased from 30 X 60 to some larger values. The effect of this will be to make the far wake boundary conditions used in the program more appropriate the larger the region size, the better these conditions become. However, execution time will increase with such an enlargement. Care must be taken with regard to the initial flowfield which is input to the program. The stepsizes used must be compatible with the following criterion:

1. There must be at least three or four points within the boundary-layer. Therefore ΔY must be appropriatly small.
2. The ratio $\Delta Y/\Delta X$ must match the slope of the body at the tail region. Since the full Navier-Stokes equations are solved, the solution is very much dependent on the time stepsize. For high velocities a very small time differential (0.1×10^{-4} sec) must be used.

Numerical Procedure

1. Set boundary conditions for $\xi^{(n+1)}$ and $\psi^{(n+1)}$. Coefficients for $u^{(n+\frac{1}{2})}$ and $v^{(n+\frac{1}{2})}$ are extrapolated from the previous time step. Boundary conditions are known except for $\xi_w^{(n+1)}$ which must be iterated upon.
2. Solve the Navier-Stokes equation using an Alternating Difference Implicit (ADI) scheme to get $\xi^{(n+1)}$.
3. Solve for $\psi^{(n+1)}$ using a direct solver method with $\xi^{(n+1)}$ for the right side.
4. Calculate new values of $\psi^{(n+1)}$ and $\xi^{(n+1)}$.
5. Iterate on ψ and ξ until $\partial u/\partial y)_w - \partial u/\partial y$ meets the convergence criterion.
6. Compute $u^{(n+1)}$ and $v^{(n+1)}$ from the new stream function.
7. Modify the stepsize, if necessary, to aid convergence.

Figures 10 and 11 show the flow of logic and the calling sequence of subroutines in program APLNS.

See Appendix A, Table 5.1 for a listing of program APLNS.

Description of Variables in Common.

BLOCK	VARIABLE	DESCRIPTION OF USE
REALS	DFRODR	$1/\rho\ \partial Fx/\partial r$ lb-ft^2/slug
REALS	DT	Δt; time increment (sec)
REALS	DTA	Δt; time increment (sec)
REALS	DT2	$\Delta t/2$
REALS	DX	Axial length increment (ft)

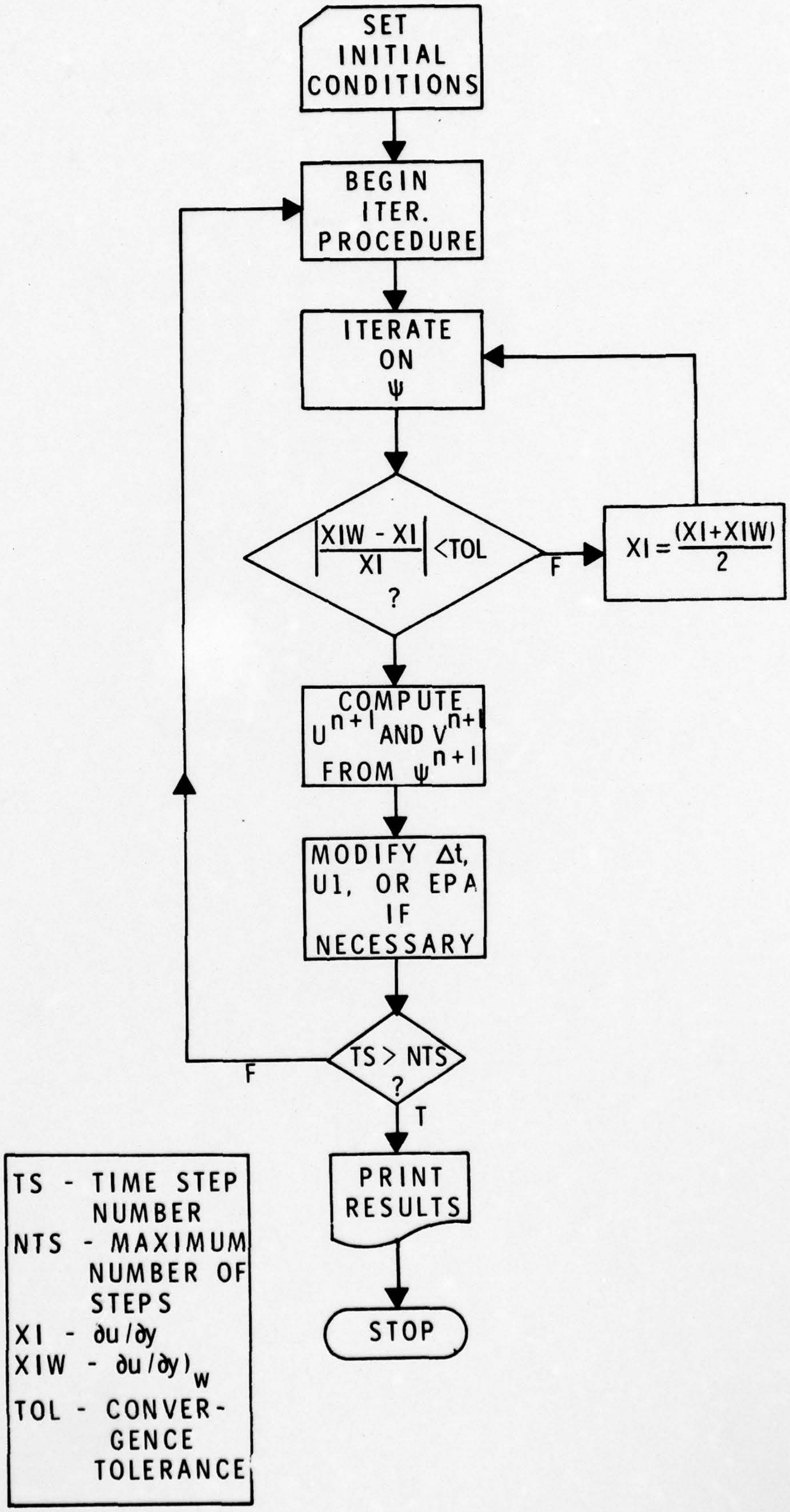


Figure 10. PROCEDURE FLOWCHART OF APLNS

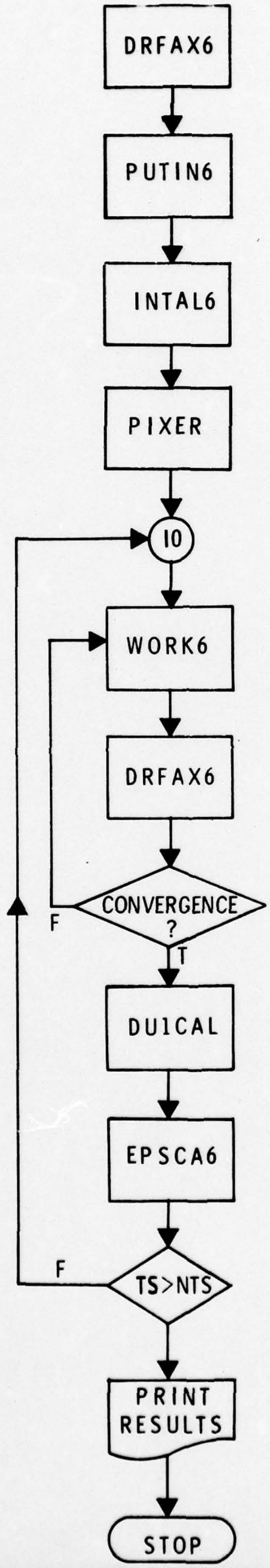


Figure 11. ROUTINE SEQUENCE FLOWCHART OF APLNS

REALS	DY	Radial length increment (ft)
REALS	DY2	$(\Delta y)^2$ $(ft)^2$
ARRAYS	EPA	ε, epsilon at each axial lication (eddy viscosity)
REALS	EPO	Beginning value of epsilon (ft^2/sec)
REALS	EPS	Value of epsilon, eddy viscosity, if ≤ 0 $\varepsilon = 0.0168 \delta U_e (ft^2/sec)$
REALS	FX	Fx; thrust magnitude (lb)
REALS	FXRHO	Fx/ρ ($lb\text{-}ft^3/slug$)
INTEGS	IDIMY	Number of columns for DMT010
INTEGS	IO	Input-output unit for any saved data
INTEGS	JCI	Point number on wall at the front of the region
INTEGS	JDEL	Boundary-layer edge point at the front of the region
INTEGS	JDIMY	Number of rows for DMT010
INTEGS	L	Number of points per profile
IRRAYS	LAR	Array which defines body shape
LOGICS	LSWH1	Control parameter set to FALSE value
INTEGS	L1	Height of the propeller zone in number of points

INTEGS	M	Number of profiles
INTEGS	M1	Number of profiles on the body
INTEGS	NER	Number of iterations on WORK6
INTEGS	NTS	Number of iterations for solution convergence
LOGICS	OFF	FALSE
LOGICS	ON	TRUE
LOGICS	PRNT	Controls printing for saved output
ARRAYS	PSI	ψ; stream function (ft^3/sec)
ARRAYS	PSIB	ψ; used in WORK6 (ft^3/sec)
ARRAYS	PSIO	ψ; on top of region (ft^3/sec)
INTEGS	PSKP	Controls frequency of DU1CAL and EPSCA6 execution
REALS	RHO	ρ; density (slugs/ft^3)
REALS	R1	Modifies stepsize
LOGICS	SAVE	Control for saving output
PLOTZ	SKEY	Key for contour plots
PLOTZ	SL	Variables in PIXER
PLOTZ	SL1	Variables in PIXER
PLOTZ	SU	Variables in PIXER
LOGICS	SWHFX	Control for propeller
REALS	TDX	$2\Delta X$ (ft)
REALS	TDY	$2\Delta Y$ (ft)

AD-A074 972 VIRGINIA POLYTECHNIC INST AND STATE UNIV BLACKSBURG --ETC F/G 20/4
USER'S GUIDE FOR A SYSTEM OF COMPUTER PROGRAMS TO PREDICT FLOWS--ETC(U)
NOV 78 M C HYMAN, S C HOULIHAN, J A HILL

UNCLASSIFIED VPI-AERO-096 NL

2 OF 4

ADA 074972

2 OF 4

ADA 074972

REALS	TOL	Tolerence for convergence of WORK6
REALS	TRAMP	Modification factor of $1/\rho\ (\partial Fx/\partial r)$ if necessary
ARRAYS	U	Axial velocity (ft/sec)
ARRAYS	UA	Axial velocity in WORK6 (ft/sec)
REALS	U0	Boundary-layer velocities at the front of the region (ft/sec)
ARRAYS	U1	Axial velocity at upper boundary of the region (ft/sec)
REALS	U2	U_e; edge velocity (ft/sec)
ARRAYS	U10	Initial axial velocity at upper boundary of the region (ft/sec)
ARRAYS	V	Radial velocity (ft/sec)
ARRAYS	VA	Radial velocity in WORK6 (ft/sec)
REALS	X	Axial location from front of the region (ft)
ARRAYS	XI	$\partial u/\partial y$ (sec^{-1})
ARRAYS	XIB	$\partial u/\partial y$ calculation at the wall from stream function in WORK6 (sec^{-1})
ARRAYS	XIW	$\partial u/\partial y$ at the wall (sec^{-1})
REALS	Y	Radial distance from body centerline (ft)

Description of Subroutines.

DRFAX6. DRFAX6 is the main driver for the axisymmetric case. The routine calls the major routines and tests for convergence of WORK6 calculations. The routine first calls two subroutines which read in data and set up initial conditions. Next, the routine begins the iteration procedure on the stream function, the main iteration loop of the program. After the convergence is met, the u velocities on the upper boundary are modified every PSKP time steps and modifications to the eddy viscosity are made every PSKP+1 time steps. The iteration on WORK6 is repeated NTS time steps with Dt modified depending on the number of iterations required for convergence of the previous time step. If more than three iterations were required, Dt is decreased by R1. If less than three iterations are required, the time step Dt is increased by the same factor R1. After NTS time steps or the number of iterations in WORK6 exceeds NER, DRFAX6 calls a routine to print the current stream function, $\partial u/\partial y$ distribution, and the u and v velocity components.

Subroutine DU1CAL. DU1CAL calculates a Δu that is added to the u velocity on the upper boundary of the region. The routine is called periodically to update this boundary condition. A modified value for the stream function is used to calculate the Δu. The routine then prints out the profile number (I), the modified upper boundary condition (UI), the difference between the current velocity and the initial velocity (DDU1), the initial velocity (U10), and the initial stream function value (PSIO). It is important to note that only velocities on the upper boundary are altered in this routine.

Subroutine EPSCA6. EPSCA6 evaluates the eddy viscosity law. It is called periodically as DU1CAL is called. The equation used is:

$$\frac{d\varepsilon(x)}{dx} = \frac{C_2^2 U_e}{2} \left[\frac{I_2}{I_1}\right] - \frac{a_2 \varepsilon^2}{4U_e C_2 b^2 I_1} - \frac{\varepsilon_1}{2I_1}\frac{dI_1}{dx} + \frac{\varepsilon}{2b}\frac{db}{dx} - \frac{\varepsilon}{2U_e}\frac{dU_e}{dx}$$

where,

$$I_1(x) = \int_0^{\ell} \bar{u}\bar{r}\, dr,$$

$$I_2(x) = \int_0^{\ell} \frac{\partial \bar{u}}{\partial f}\, \bar{r}\, d\bar{r}$$

and

$$\ell = C_2 b, \quad \bar{u} = U/U_e, \quad \bar{r} = r/b, \quad a_2 = 2.0, \quad C_2 = 0.035.$$

The routine prints out the following values

I1	Defined above
I2	Defined above
B	Value of b in above equation
DB	ΔB
DI1	ΔI_1
F1, F2, F3	Terms in the eddy viscosity law
EPN	New value of $\varepsilon(x)$
EPA	Original value of $\varepsilon(x)$

These updated values are then returned to the program and used to modify the flowfield conditions.

Subroutine INTAL6. INTAL6 is responsible for calculating the inviscid stream function from flowfield velocities read in from Unit 8 data. A simple integration, which has been found to be accurate,

is used to obtain the inviscid stream function. Since the velocities at every station in the flowfield are known, the stream function is calculated using a trapazoidal rule:

$$\Delta\psi_{ij} = (u_{ij}y_i + u_{i,j-1}y_{i-2})\Delta y/2$$

$$\psi_{ij} = \psi_{i,j-1} + \Delta\psi_{i,j}$$

with $\psi = 0$ at solid body boundaries and for zero radius in the wake. The inviscid flowfield is printed out giving values of ψ(PSI) and u and v velocity components. The value of r at the edge of the boundary-layer at the front of the region (Y(JDEL)) is printed along with the boundary-layer thickness (Y(JDEL)-Y(J(I))), and U_e (U2). The boundary-layer velocity profiles are read in from Unit 11, and the values at the point in question Y(J) are interpolated using a 3 point Lagrangian formula. The momentum increase (FX(1,L1)) is printed out giving the initial values of thrust computed from the following equation;

$$Fx/\rho = Fx_0 = 2\frac{SU_0}{y(L1)^2 DX} \text{ ; where } SU_0 = \int_0^\delta U(U_e - U)dy$$

In attempting to correct for tip losses the following expression was used:

$$Fx/\rho = \frac{Fx_0}{2}\left\{1 + \cos\left[\pi \frac{r-(L1-5)\Delta r}{4\Delta r}\right]\right\} \text{ for } (L1-5)\Delta r<r<(L1-1)\Delta r$$

where L1(dr) is the height of the propeller region. The value of δ^{*2} (DELSR2) is calculated by;

$$\delta^{*2} = \int_0^\delta (1 - \frac{U}{U_e})rdy$$

and the corresponding value of ε(EPS) is given by: $\varepsilon = 0.0168\delta^* U_e$. The boundary-layer velocity profile (UO) is printed out along with the values of X and Y for the region in question. The X distances are taken from the front of the region and the Y values are measured from the axis of the body. Finally $(1/\rho)\ \partial F_x/\partial r$(FXRHO) is printed completing the printed output from the routine.

Subroutine PUTIN6. PUTIN6 reads in values for the flags and various other parameters in the program and prints them out in namelist form. The routine defines the body shape by the LAR array which is also printed out. The initial values for variables which control the key values for the contour plot of $\partial u/\partial y$(XI) are printed. For a description of these parameters see the section entitled Description of Input.

Subroutine WORK6. WORK6 calculates the updated values of ξ which are compared in DRFAX6 and then modified if necessary. Since this routine essentially solves n equations in n unknowns, only the numerical procedure is important to the understanding of the workings of WORK6. The use of WORK6 is best seen by referring to the section entitled Numerical Procedure, steps 1-5.

Description of Input.

Input Variables on Unit 5.

Namelist	Variable		Description of Use
ILIST	L	,	Number of points per profile, suggested value ≤30
ILIST	L1	,	Number of points on propeller, suggested value ≤12
ILIST	M	,	Number of profiles, suggested value ≤60

ILIST	M1	,	Number of profiles on body, suggested value ≤15
ILIST	PSKP	,	Time parameter controlling frequency of DU1CAL and EPSCA6, suggested value 4-7
ILIST	NER	,	Number of iterations on WORK6, suggested value 40
ILIST	NTS	,	Number of time steps for convergence, suggested value 35
ILIST	IO	,	Unit number for saved output, suggested value 9
RLIST	DT	,	Time increment (sec), suggested value 1×10^{-4}-1×10^{-5}
RLIST	RHO	,	Density of fluid (slugs/ft^3) suggested value 7.535×10^{-2}
RLIST	EPS	,	Eddy viscosity (if EPS≤0, EPS = $0.0168\delta^* U_e$), suggested value 0.0
RLIST	TOL	,	Tolerance value for convergence of WORK6, suggested value 0.02-0.05
RLIST	TRAMP	,	Modifying factor for propeller, suggested value 0.1
RLIST	EPO	,	Beginning value of epsilon, suggested value 0.0
RLIST	SCALE1	,	Length scaling factor for nondimensional HESS/ICBLINT I/O
RLIST	SCALE2	,	Velocity scaling factor for nondimensional HESS/ICBLINT I/O
LLIST	SWHFX	,	T No actuator disk F Include actuator disk

LLIST	SAVE	,	T Save data on previous unit defined by variable IO F Do not save data

See Appendix A, Table 3.1 for sample APLNS data.

Input Data on Unit 11.

Variable	Description of Variable
DYDX	dy/dx; slope of the body at the tail
DY	Δy; nondimensional radial lingth increment
NUM	Number of boundary-layer velocities given
DELTA	Nondimensional boundary-layer thickness
UE	Nondimensional edge velocity
YD	y locations in the boundary layer
UUE	U/U_e; normalized boundary-layer velocities

See Appendix B, Table 3.2 for sample APLNS Unit 11 input data.

Input Data on Unit 8.

X	Nondimensional axial distance from the front of the body
Y	Nondimensional radial coordinate from the body centerline
U	Nondimensional axial velocity component (INVISCID)
V	Nondimensional radial velocity component (INVISCID)

See Appendix B, Table 3.3 for sample APLNS Unit 8 data.

Description of Output.

Output Variables. Listed in order of appearance.

Origin	Variable	Description of Use
PUTIN6	L	Number of points per profile

PUTIN6	L1	Number of points in the propeller zone (height)
PUTIN6	M	Number of profiles
PUTIN6	M1	Number of profiles on the body
PUTIN6	PSKP	Control variable concerning frequency of DU1CAL and EPSCA6
PUTIN6	NER	Number of iterations on WORK6 (maximum)
PUTIN6	NTS	Total number of time steps for solution convergence
PUTIN6	IO	Unit number for saved output
PUTIN6	DT	Time increment (s)
PUTIN6	RHO	Density (slugs/ft^3)
PUTIN6	EPS	Eddy viscosity (ft^2/s)
PUTIN6	TOL	Tolerance value for convergence of WORK6
PUTIN6	TRAMP	Modification factor for derivative of thrust in the radial direction
PUTIN6	EPO	Eddy viscosity at beginning of problem (ft^2/s)
PUTIN6	SCALE1	Length scaling factor (ft)
PUTIN6	SCALE2	Velocity scaling factor (ft/s)
PUTIN6	SWHFX	Control parameter for propeller
PUTIN6	SAVE	Control parameter for printing saved output
PUTIN6	LAR	Array defining body points in terms of grid location

PUTIN6	DX	Axial length increment(ft)
PUTIN6	DY	Radial length increment (ft)
PUTIN6	JDEL	Point number at the edge of the boundary-layer
INTAL6*	PSI	Inviscid stream function (ft^3/s)
INTAL6*	U	Inviscid axial velocity (ft/s)
INTAL6*	V	Inviscid radial velocity (ft/s)
INTAL6	Y(JDEL)	Radial distance at the edge of the boundary-layer (ft)
INTAL6	Y(JDEL)-Y(J(I))	Boundary-layer thickness (ft)
INTAL6	U2	Axial velocity at the edge of the boundary-layer (ft/s)
INTAL6	FX	Force distribution of the actuator zone (lb_f)
INTAL6	DELSR2	Square of the boudary-layer displacement thickness (ft^2)
INTAL6	EPS	Eddy viscosity (ft^2/s)
INTAL6	UO	Boundary-layer profile (ft/s)
INTAL6	Y	Radial distances of grid points from body centerline (ft)
INTAL6	X	Axial distances of grid points from front of region (ft)
INTAL6	FXRHO	Derivative of propeller force with respect to radial direction (lb_f-ft^3/slug)

DRFAX6	TIME	Time from the beginning of the procedure (s)
DRFAX6	XI	Derivative of axial velocity with respect to radial direction (1/s)
WORK6	N	Dimension of working matrix in WORK6
WORK6	IDIMY	Dimension of working matrix in WORK6
WORK6	JDIMY	Dimension of working matrix in WORK6
WORK6	I	Profile number
WORK6	J	Point number on profile
WORK6	XI(I,J)	XI at point I,J (1/s)
WORK6	X1W	Vorticity at the wall of the body at point I,J (1/s)
DRFAX6	ITER	Iteration number in WORK6
DRFAX6	XMAX	Maximum difference between $\lvert(XI-X1W)/XI\rvert$ for all wall points
DRFAX6	TS	Time step number
DU1CAL	U1(I)	Updated axial velocity at the top of the region (ft/s)
DU1CAL	DDU1(I)	Difference between updated and original upper boundary axial velocity (ft/s)
DU1CAL	U10(I)	Initial axial velocity at the upper boundary (ft/s)
DU1CAL	PSI0	Stream function at the top of the boundary (ft^3/s)
EPSCA6	B	Mixing zone width (ft)
EPSCA6	I1	$I_1(x) = \int_0^{\ell} \bar{u}\bar{r}\, d\bar{r}$

EPSCA6	I2	$I_2(x) = \int_0^{\ell} \frac{\partial \bar{u}}{\partial \bar{r}} \bar{r}\, d\bar{r}$
EPSCA6	DB	Change in the value of the mixing width (ft)
EPSCA6	DI1	Change in the value of $I_1(x)$ integral
EPSCA6	F1	Term in the eddy viscosity law
EPSCA6	F2	Term in the eddy viscosity law
EPSCA6	F3	Term in the eddy viscosity law
EPSCA6	EPN	New value of the eddy viscosity at profile I (ft^2/s)
EPSCA6	EPA	Initial value of the eddy viscosity for profile I (ft^2/s)
DRFAX6*	PSI	Final stream function (ft^3/s)
DRFAX6*	XI	Final distribution for XI (1/s)
DRFAX6	V	Final values of radial velocities (ft/s)
DRFAX6	U	Final values of axial velocities (ft/s)

* - Denotes p[illegible]nted within mentioned subroutine through a call to routine DMT010

See Appendix B, Table 3.4 for sample APLNS OUTPUT data.

APPENDIX A

APPENDIX A

TABLE 1.1 WAKE EXEC

```
CP SPOOL CONS TO SYSTEM CLASS A
CP TAG DEV CONS PRIOR IDLE DEST RMT35
CP TAG DEV PR PRIOR IDLE DEST RMT35
CP LINK GOOD 192 192 RR ALL
ACCESS 192 D
EXEC CDCP 5
EXEC BLINT 6
EXEC DEVELOP 5
```

TABLE 1.2 CDCP EXEC

```
&LOOP -ENDLOOP &1
EXEC DATADEV
EXEC HESSBLI HESSBLI DATA E
EXEC DATAMA
EXSERV READ SEPARA DATA E 3 3($COL 1 1 ICP
&IF .&ICP NE .1 &GOTO -NEXT
-ENDLOOP &CONTINUE
&TYPE MAX. NO. OF ITER. EXCEEDED.
```

```
&EXIT 8
-NEXT &CONTINUE
```

TABLE 2.2 DATADEV EXEC

```
&CONTROL ALL
FI * CLEAR
GLOBAL TXTLIB FORTLIB VPIUTIL
FI 01 DISK COORD DATA E (LRECL 80 BLKSIZE 80 RECFM FB
FI 02 DISK SEPARA DATA E (LRECL 80 BLKSIZE 80 RECFM FB
FI 03 DISK WAKE COORDS E (LRECL 80 BLKSIZE 80 RECFM FB
FI 04 DISK HESS PARMDATA E (LRECL 80 BLKSIZE 80 RECFM FB
FI 07 DISK HESSBLI DATA E (LRECL 80 BLKSIZE 80 RECFM FB
FI 06 DISK ERERR DAATAA A
LOAD DATADEV (NOMAP
START
```

TABLE 2.3 DATADEV FORTRAN

```
C     PROGRAM MAIN
C
C
```

```
C
C                          FILE DEFINITIONS
C
C
C                  UNIT (1)    BODY COORDINATES WITH PARAMETERS
C                  UNIT (2)    FLOW SEPARATION AND ITERATION DATA
C                  UNIT (4)    POTENTIAL FLOW CODE EXECUTION PARAMETERS
C                  UNIT (7)    COMPLETE DATA SET FOR POTENTIAL FLOW CODE
C                              EXECUTION WITH FAIRED AFTERBODY.
C
C
C
C*********************************************************************
C
C
C
      COMMON /COORD/ X,Y,KTE,XI,YI,KTE1,NOPRFS,KWAKE
      DIMENSION X(501), Y(501), XI(501), YI(501)
C
C     READ IN THE ORIGINAL BODY
C
      READ (1,30) KWAKE
      READ (1,10)  KTE, NOPRFS, KTE1
      DO 1 I=1,KTE
    1 READ (1,20) X(I), Y(I)
      REWIND 1
      CALL LINDEV
      CALL HESSPA
   10 FORMAT (1X,3I3)
   20 FORMAT (2F12.6)
   30 FORMAT (I3)
      STOP
      END
```

```
      SUBROUTINE LINDEV
      COMMON /COORD/ X,Y,KTE,XI,YI,KTE1,NOPRFS,KWAKE
      COMMON /CORFIT/DX,Y1,C,X1,NNN
      DIMENSION X(501),Y(501),XI(501),YI(501),XWAKE(20),XXI(500),YYI(500
     1)
C
C     READ IN SEPARATION PARAMETERS
C
      READ (2,10) IWAKE
      READ (2,10) JSTA
      KSTA = JSTA-IWAKE
C
C     COPY ORINGINAL COORDINATES TO WORKING ARRAYS
C
      DO 1 I=1,KTE
      XI(I) = X(I)
    1 YI(I) = Y(I)
      IF(IWAKE.EQ.0) GO TO 9
C
C      CALCULATE REPOSITIONED X COORDINATES
C
      DX = .01*XI(KTE)
      N = 0
      XX = ROUND(XI(KSTA),2)
    2 N = N+1
      XXI(N) = XX+DX*(N-1)
      IF((XI(KTE)-XXI(N)).GT..0001) GO TO 2
      KK = N
C
C        CALCULATE ACCOMPANYING Y COORDINATES
C
      DO 3 I=1,KK
```

```
      DO 3 J=KK,KTE
      IF((XI(J).LT.XXI(I)).AND.(XI(J+1).GT.XXI(I))) YYI(I) =-((XXI(I)-XI
     1(J))/(XI(J+1)-XI(J)))*(YI(J)-YI(J+1))+YI(J)
      IF(XI(J).EQ.XXI(I)) YYI(I) = YI(J)
    3 CONTINUE
      DO 4 I=1,KTE
      IF((XI(I).LE.XXI(1)).AND.(XI(I+1).GT.XXI(1))) GO TO 6
    4 CONTINUE
C
C        REPLACE ORIGINAL WITH RELOCATED COORDINATES, MM IS THE TOTAL
C        NUMBER OF NEW COORDINATES ON THE BODY.
C
    6 NN = I
      NNN = NN+1
      MM =NN+KK
      JJ = 0
      DO 7 I=NNN,MM
      JJ = JJ+1
      XI(I) = XXI(JJ)
    7 YI(I) = YYI(JJ)
      KTE = MM
    5 CONTINUE
C
C        DEFINE THE NUMBER OF POINTS IN WAKE EXTENSION
C
      LWAKE = IWAKE*5
      KTE1 = KTE+LWAKE
    9 CONTINUE
C
C       WRITE OUT BODY COORDINATES FOR USE ON NEXT ITERATION IF ONE
C
      WRITE (1,15) KWAKE
```

```
      WRITE(1,10)  KTE, NOPRFS,KTE1
      WRITE (1,40) (XI(I),YI(I),I=1,KTE)
      IF(IWAKE.EQ.0) GO TO 13
      DO 8 I=KTE,KTE1
    8 XI(I) = XI(I-1)+DX
C
C        DEVELOP DATA FOR CURVE FIT
C
      X1 = XI(KTE1)-XI(NNN)
      CALL FD3(XI(NNN),XI(NNN-1),XI(NNN),XI(NNN+1),YI(NNN-1),YI(NNN
     1),YI(NNN+1),FX)
      C = FX
      Y1 = YI(NNN)
      CALL FIT
   13 REWIND 2
      WRITE (2,10) IWAKE
      WRITE (2,10) JSTA,KTE
   10 FORMAT (1X,3I3)
   15 FORMAT (I3)
   20 FORMAT (F12.6)
   40 FORMAT (2F12.6)
      DEBUG INIT
      RETURN
      END
```

```
      SUBROUTINE HESSPA
C
C        SUBROUTINE HESSPA WRITES OUT A HESS DATA FILE USING COORDINATES
C        DEVELOPED BY FIT
C
      COMMON /COORD/ X,Y,KTE,XI,YI,KTE1,NOPRFS,KWAKE
      DIMENSION X(501),Y(501),XI(501),YI(501)
      DIMENSION HEDR(15),CASE(2)
      READ (4,10) HEDR,CASE,ICARD2,ICARD3,ICARD4,NN,IBDN
      WRITE(7,10) HEDR,CASE,ICARD2,ICARD3,ICARD4,KTE1,IBDN
      DO 1 J=1,KTE1
    1 WRITE (7,20) XI(J),YI(J)
   10 FORMAT (15A4,2X,2A4/I3/I5/1X,I4,I5/9X,I1)
   20 FORMAT (2F12.6)
      RETURN
      END
```

```
      SUBROUTINE FIT
C
C       FIT DEVELOPS A NUMERICALLY GENERATED CURNE GIVEN THE COORDINATES
C       AND SLOPE AT TWO POINTS (AT THE T.E. THE SLOPE IS ASSUMED 0)
C
      REAL K,M
      COMMON/CORFIT/ DX,Y1,C,X1,NNN
      COMMON /COORD/ X,Y,KTE,XI,YI,KTE1,NOPRFS,KWAKE
      DIMENSION X(501),Y(501),XI(501),YI(501),XXI(500),YYI(500)
      M = C
C
C     DELY IS THE ACTUAL Y STEP SIZE, DY IS THE INCREMENT OF DELY FOR
C     THE NEXT ITERATION
C
      DELY = 0.
      DELX = DX
      DY = -.05*DELX
   30 DELY = DELY+DY
      XPRIME = 0.
      YPRIME = 0.
      N = X1/DELX
      K = M*DELX/DELY
      L = NNN
C
C      GENERATE Y'S FOR CURRENT ITERATION
C
      MF = N+1
      DO 5 I=1,MF
      L = L+1
      YPRIME = YPRIME+DELY*K**I
    5 Y(L)=Y1+YPRIME
C
```

```
C      CHECK FOR CONVERGENCE
C
C
C      IF UNDERSHOOT, CONTINUE
C
       IF(ABS(Y(KTE1)).LT..00001) GO TO 20
C
C       IF OVERSHOOT, GO BACK AND USE SMALLER DY'S
C
       IF(Y(KTE1).LT.0.) GO TO 30
       DELY=DELY-DY
       DY = DY*.1
       GO TO 30
    20 CONTINUE
       DO 10 L=NNN,KTE1
    10 YI(L)=Y(L)
       RETURN
       END
```

```
      FUNCTION ROUND (X,N)
C     THIS FUNCTICN ROUNDS NUMBERS TO THE DESIRED DECIMAL PLACE N
      ISIGN=1
      IF (X.LT.0.0) ISIGN=-1
      I=ABS(X)*10**N+0.5
      ROUND=FLOAT(ISIGN*I)*10.0**(-N)
      RETURN
      END
```

```
      SUBROUTINE FD3 (X,X1,X2,X3,F1,F2,F3,FX)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE FD3 IS CALLED BY SUBROUTINE DERIV3.
C
C     SUBROUTINE FD3 CALCULATES THE FIRST DERIVATIVE-FX-CORRESPONDING
C     TO POINT X USING 3 POINT LAGRANGIAN DIFFERENTIATION FORMULA.
C
C     ASSUMES X1 .LE. X .LE. X3.
C
      A1=2.0*X-X2-X3
      A2=2.0*X-X1-X3
      A3=2.0*X-X1-X2
      D1=(X1-X2)*(X1-X3)
      D2=(X2-X1)*(X2-X3)
      D3=(X3-X1)*(X3-X2)
      C1=A1/D1
      C2=A2/D2
      C3=A3/D3
      FX=C1*F1+C2*F2+C3*F3
      RETURN
         END
```

TABLE 2.3 DATAMA EXEC

```
&CONTROL ALL
FI * CLEAR
GLOBAL TXTLIB FORTLIB VPIUTIL
FI 03 DISK WAKE COORDS E (LRECL 80 BLKSIZE 80 RECFM F
FI 04 DISK ICBL PARMDATA E (LRECL 100 BLKSIZE 80 RECFM FB
FI 05 DISK HESSBLI DATA E (LRECL 80 BLKSIZE 80 RECFM FB
FI 06 DISK SUB DATA E (LRECL 100 BLKSIZE 80 RECFM FB
FI 07 DISK SEPARA DATA E (LRECL 80 BLKSIZE 80 RECFM FB
FI 08 DISK SUMDS DATA E (LRECL 80 BLKSIZE 80 RECFM FB
FI 09 DISK HESSBLI OUTPUT E (LRECL 80 BLKSIZE 80 RECFM FB
FI 10 DISK HESSCP OUTPUT E (LRECL 80 BLKSIZE 80 RECFM FB
FI 11 DISK COORD DATA E (LRECL 80 BLKSIZE 80 RECFM FB)
FI 12 DISK INTER DATA B
LOAD DATAMA (NOMAP
START
```

TABLE 2.5 DATAMA FORTRAN

```
C                                                                       MAI00010
C                                                                       MAI00020
```

```
C                                                                       MAI00030
C*********************************************                          MAI00040
C                                                                       MAI00050
C                                                                       MAI00060
C                                                                       MAI00070
C                      FILE DEFINITIONS                                 MAI00080
C                                                                       MAI00090
C                                                                       MAI00100
C                                                                       MAI00110
C               UNIT (3)        WAKE COORDINATES                        MAI00120
C               UNIT (4)        BOUNDARY LAYER CODE EXECUTION PARMETERS MAI00130
C               UNIT (5)        POTENTIAL FLOW CODE DATA SET WITH FULL  MAI00140
C                               BODY COORDINATE INFORMATION.            MAI00150
C               UNIT (6)        SUB DATA (FINAL VERSION WITH WAKE POINTS) MAI00160
C               UNIT (7)        DATA SET FOR DEVELOPING NONSEPARATING   MAI00170
C                               AFTERBODY                               MAI00180
C               UNIT (8)        SUMDS DATA (FROM HESSBLI CODE)          MAI00190
C               UNIT (9)        HESSBLI OUTPUT (FROM HESSBLI CODE)      MAI00200
C               UNIT (10)       HESSCP OUTPUT (FOR ICBLINT CODE)        MAI00210
C               UNIT (11)       ORIGINAL COORDINATES WITH AFTERBODY     MAI00220
C               UNIT (12)       DATA FOR DEVELOPING THE INVISCID FLOW   MAI00230
C                               FIELD USING THE DISPLACED BODY          MAI00240
C               UNIT (13)       DATA FOR RUNNING THE C.P. INTERPOLATION MAI0025C
C                               PROGRAM; BLOMKER.                       MAI00260
C                                                                       MAI00270
C                                                                       MAI00280
C*********************************************                          MAI00290
      COMMON /WAKE/ XWAKE,YWAKE,KTE,IIMAX,KWAKE,KTE1,X,Y,XI,YI,JSTA     MAI00300
     1,DYDX, XLOCAT,JWAKE                                               MAI00310
      DIMENSION XWAKE(20), YWAKE(20), X(501), Y(501), XI(501), YI(501)  MAI00320
C                                                                       MAI00330
C        READ IN WAKE COORDINATES IF USED                               MAI00340
```

```
C                                                                       MAI00350
      READ (11,30) KWAKE                                                MAI00360
      READ(3,10) JWAKE                                                  MAI00370
      DO 1 I=1,JWAKE                                                    MAI00380
    1 READ(3,20) XWAKE(I)                                               MAI00390
      DO 2 J=1, JWAKE                                                   MAI00400
    2 YWAKE(J) = 0.0                                                    MAI00410
      CALL CPDATA                                                       MAI00420
C                                                                       MAI00430
C     IF THE FAIRING DOES NOT DRIVE THE CP'S LOW ENOUGH DO NOT WRITE    MAI00440
C     OUT HESS/ICBLINT DATA SETS                                        MAI00450
C                                                                       MAI00460
C     IF(ICP.NE.0) GO TO 3                                              MAI00470
      CALL CURV                                                         MAI00480
      CALL READAT                                                       MAI00490
   10 FORMAT (I2)                                                       MAI00500
   20 FORMAT (F12.6)                                                    MAI00510
   30 FORMAT(I3)                                                        MAI00520
    3 STOP                                                              MAI00530
      END                                                               MAI00540
```

```
      SUBROUTINE CPDATA                                                 CPD00010
C                                                                       CPD00020
C      CPDATA DEVELOPS AN ON-COORDINATE CP DATA SET AND DECIDES IF      CPD00030
C      FAIRING DROVE THE CP'S SUFFICIENTLY LOW                          CPD00040
C                                                                       CPD00050
      COMMON /WAKE/ XWAKE,YWAKE,KTE,IIMAX,KWAKE,KTE1,X,Y,XI,YI,JSTA     CPD00060
     1,DYDX,XLOCAT,JWAKE                                                CPD00070
      DIMENSION XWAKE(20), YWAKE(20), X(501), XW(501), CP(501), F(501)  CPD00080
      DIMENSION XI(501),YI(501)                                         CPD00090
      DIMENSION Y(501),XTO(501)                                         CPD00100
C                                                                       CPD00110
C      READ IN THE SEPARATION DATA                                      CPD00120
C                                                                       CPD00130
      READ (7,20) IWAKE                                                 CPD00140
      READ (7,20) JSTA,KTE                                              CPD00150
      LWAKE = IWAKE*5                                                   CPD00160
      IIMAX = LWAKE+KTE                                                 CPD00170
    5 KTE1 = KTE+LWAKE                                                  CPD00180
C                                                                       CPD00190
C       READ IN OFF-COORDINATE CP VALUES                                CPD00200
C                                                                       CPD00210
      READ(9,10) (XW(I), XTO(I), F(I), I=1,KTE1)                        CPD00220
      KMINUS = KTE1-1                                                   CPD00230
      CP(1) = F(1)                                                      CPD00240
C                                                                       CPD00250
C      ITERPOLATE OFF-COORDINATE CP'S TO ON-COORDINATE CP'S             CPD00260
C                                                                       CPD00270
      DO 1 K = 2, KMINUS                                                CPD00280
    1 CP(K) = (F(K)+F(K+1))/2.0                                         CPD00290
      DELX1 = XTO(KTE1)-XTO(KMINUS)                                     CPD00300
      DELX2 = XW(KTE1)-XW(KMINUS)                                       CPD00310
      DEL = DELX2/DELX1                                                 CPD00320
```

```
      CP(KTE1) = (1.+DEL)*F(KTE1)-DEL*F(KMINUS)                         CPD00330
C                                                                       CPD00340
C      FIND LAST, MOST NEGATIVE CP'S AND LOCATION FOR USE IN GIVING INITCPD00350
C      INITIAL WAKE PRESSURES.  ALSO DETERMINE IF CP'S GO TO HIGH IN    CPD00360
C      REGION OF TRAILING EDGE                                          CPD00370
C                                                                       CPD00380
      INFL = 0                                                          CPD00390
      ICP = 0                                                           CPD00400
      DO 2 I= 2,KTE1                                                    CPD00410
      IF (((CP(I-1).GT.CP(I)).AND.(CP(I+1).GT.CP(I)).AND.(CP(I).LE.0.)))CPD00420
     1ICPAR = I                                                         CPD00430
      IF(((CP(I).GT.0.).AND.(CP(I-1).LT.0.)).OR.((CP(I).LT.0.).AND.     CPD00440
     1(CP(I-1).GT.0.))) INFL = INFL+1                                   CPD00450
      IF((INFL.EQ.2).AND.(CP(I).GE.0.2)) ICP = 1                        CPD00460
      IF(((CP(I).GT.0.).AND.(CP(I-1).LT.0.)).OR.((CP(I).LT.0.).AND.     CPD00470
     1(CP(I-1).GT.0.))) JSTA = I                                        CPD00480
    2 CONTINUE                                                          CPD00490
      IF(KWAKE.EQ.0) GO TO 6                                            CPD00500
C                                                                       CPD00510
C        FIND WHERE WAKE STARTS AND FAIRING ENDS                        CPD00520
C                                                                       CPD00530
      JTE = KTE1+1                                                      CPD00540
      DO 8 I=1,JWAKE                                                    CPD00550
      IF (XW(KTE1).LT.XWAKE(I)) GO TO 9                                 CPD00560
    8 CONTINUE                                                          CPD00570
      GO TO 6                                                           CPD00580
C                                                                       CPD00590
C        PUT WAKE POINTS INTO ACTUAL COORDINATE ARRAYS                  CPD00600
C                                                                       CPD00610
    9 II = I                                                            CPD00620
      IIMAX = KTE1+JWAKE-II                                             CPD00630
      ICOUNT = II                                                       CPD00640
```

```
      DO 3 I=JTE,IIMAX                                                  CPD00650
      ICOUNT = ICOUNT+1                                                 CPD00660
      XW(I) = XWAKE(ICOUNT)                                             CPD00670
    3 CP(I) = CP(ICPAR)                                                 CPD00680
    6 DO 4 I=1,IIMAX                                                    CPD00690
C                                                                       CPD00700
C       WRITE OUT THE FULL ON-COORDINATE CP DATA SET WITH WAKE POINTS IFCPD00710
C       ANY                                                             CPD00720
    4 WRITE(10,10) XW(I), CP(I)                                         CPD00730
      IWAKE = IWAKE+1                                                   CPD00740
      REWIND 7                                                          CPD00750
C                                                                       CPD00760
C       WRITE OUT THE SEPARATION DATA                                   CPD00770
C                                                                       CPD00780
      WRITE (7,20) IWAKE                                                CPD00790
      WRITE (7,20) JSTA,KTE                                             CPD00800
      WRITE (7,30) ICP                                                  CPD00810
      RETURN                                                            CPD00820
   10 FORMAT (3F12.6)                                                   CPD00830
   20 FORMAT (1X,3I3)                                                   CPD00840
   30 FORMAT (I3)                                                       CPD00850
      END
```

```
      SUBROUTINE CURV                                                   CUR00010
C                                                                       CUR00020
C       SUBROUTINE CURV DEVELOPS THE DATA FOR USE IN THE INVISCID FLOW  CUR00030
C       FIELD DEVELOPMENT, INCLUDING FITTING A STRAIGHT TAPERED TAIL TO CUR00040
C       THE ORIGINAL BODY.                                              CUR00050
C                                                                       CUR00060
      COMMON /WAKE/XWAKE,YWAKE,KTE,IIMAX,KWAKE,KTE1,X,Y,XI,YI,JSTA      CUR00070
     1,DYDX,XLOCAT,JWAKE                                                CUR00080
      DIMENSION X(501),Y(501),XI(501),YI(501),XWAKE(20),YWAKE(20)       CUR00090
      DATA IHESSI/1/                                                    CUR00100
C                                                                       CUR00110
C       READ IN ORIGINAL BODY COORDINATES                               CUR00120
C                                                                       CUR00130
      READ (11,10) KTEE,NOPRFS,KTEA                                     CUR00140
      READ (11,20) (XI(I),YI(I),I=1,KTEE)                               CUR00150
      N = 0                                                             CUR00160
    1 JSTA = JSTA-N                                                     CUR00170
      DX = (XI(KTE1)-XI(JSTA))/NOPRFS                                   CUR00180
      N = N+1                                                           CUR00190
      DYDX = YI(JSTA)/(XI(JSTA)-XI(KTE1))                               CUR00200
      DY = ABS(DYDX*DX)                                                 CUR00210
      IF(DY.LT..0020) GO TO 1                                           CUR00220
      XLOCAT = XI(JSTA)                                                 CUR00230
      WRITE (12,30) IHESSI,XLOCAT,DX,DY                                 CUR00240
      WRITE (13,30) KTE1
   30 FORMAT (I1/3F12.6)                                                CUR00250
   10 FORMAT (1X,3I3)                                                   CUR00260
   20 FORMAT (2F12.6)                                                   CUR00270
      RETURN                                                            CUR00280
      END
```

```
      SUBROUTINE READAT                                                 REA00010
C                                                                       REA00020
C       READAT READS AND WRITES OUT THE HESS/ICBLINT DATA SETS IF       REA00030
C       FAIRING ITERATION WAS SUCCESSFUL                                REA00040
C                                                                       REA00050
      COMMON /WAKE/ XWAKE,YWAKE,KTE,IIMAX,KWAKE,KTE1,X,Y,XI,YI,JSTA,    REA00060
     1DYDX,XLOCAT,JWAKE                                                 REA00070
      DIMENSION X(501), Y(501), XSTA(501), XWAKE(20), YWAKE(20), HEDR(15REA00080
     1), LABEL(18), IPR(101), IPRFL(101), CASE(2), XI(501), YI(501)     REA00090
      DATA LSTC/4HLAST/                                                 REA00100
      ZERO = 0.0                                                        REA00110
      READ(4,10) IREAD, IRSRT,ITER, KHESS                               REA00120
      WRITE(6,10) IREAD, IRSRT, ITER, KHESS                             REA00130
      READ (5,20) HEDR, CASE,ICARD2,ICARD3,ICARD4, NN,IBDN              REA00140
      WRITE(6,20) HEDR, CASE,ICARD2,ICARD3,ICARD4, KTE1,IBDN            REA00150
      READ (4,30) BETAS, ROMAX                                          REA00160
      WRITE(6,30) BETAS, ROMAX                                          REA00170
      READ (4,40) LABEL                                                 REA00180
      WRITE(6,40) LABEL                                                 REA00190
      READ (4,50) NOSE, LAMTRB, KVSLAW, KTRANS, KTRNSN, KONSET, KTE     REA00200
      WRITE(6,50) NOSE, LAMTRB, KVSLAW, KTRANS, KTRNSN, KONSET, KTE1    REA00210
      READ (4,10) NIT1, NIT2, NIT3, NC                                  REA00220
      WRITE(6,10) NIT1, NIT2, NIT3, NC                                  REA00230
      READ (4,60) IF, KEND, KKMAX                                       REA00240
      WRITE(6,60) IE, KEND, IIMAX                                       REA00250
      READ (4,10) KADETA, KL, IPFL, IPRINT, IWT                         REA00260
      WRITE(6,10) KADETA, KL, IPFL, IPRINT, IWT                         REA00270
      READ (4,30) CHICRT, WBAR, ATR                                     REA00280
      WRITE(6,30) CHICRT, WBAR, ATR                                     RFA00290
      READ (4,25) UFS, REINF, PFS, PSTAG, SCF, RNOSE                    REA00300
      WRITE(6,25) UFS, REINF, PFS, PSTAG, SCF, RNOSE                    REA00310
      READ (4,30) XF1,XF2, CONVRG, ADTEST                               REA00320
```

```
      WRITE(6,30) XF1,XF2, CONVRG, ADTEST                               REA00330
      READ (4,30) DX, CRNI, XKETA, ETAINF, REFLEN                       REA00340
      WRITE(6,30) DX, CRNI, XKETA, ETAINF, REFLEN                       REA00350
      READ (4,70) XJFAC, XKFAC, TVC                                     REA00360
      WRITE(6,70) XJFAC, XKFAC, TVC                                     REA00370
      READ (4,30) DXMAX                                                 REA00380
      WRITE(6,35) DXMAX,XLOCAT,DYD,JSTA,NOPRFS                          REA00390
      IF (IIMAX.GT.101) GO TO 4                                         REA00400
      DO 5 J=1,IIMAX                                                    REA00410
      IPR(J) = I                                                        REA00420
      IPRFL(J) = I                                                      REA00430
    5 CONTINUE                                                          REA00440
      GO TO 6                                                           REA00450
    4 IPRINT = 101                                                      REA00460
      IPFL = 101                                                        REA00470
      K = IIMAX-101                                                     REA00480
      DO 7 J=1,101                                                      REA00490
      IPR(J) = IPR(J-1)+1                                               REA00500
      IF(J.LE.K) IPR(J) = 2*J-1                                         REA00510
      IPRFL(J) = IPR(J)                                                 REA00520
    7 CONTINUE                                                          REA00530
    6 WRITE(6,80) (IPR(J),J=1,IPRINT)                                   REA00540
      WRITE(6,80) (IPRFL(J),J=1,IPFL)                                   REA00550
      READ (8,100) (XSTA(I), I=1,KTE1)                                  REA00560
      DO 2 I=1,KTE1                                                     REA00570
    2 READ (5,30) X(I), Y(I)                                            REA00580
      IF(KWAKE.EQ.0) GO TO 3                                            REA00590
      KPLUS = KTE1+1                                                    REA00600
      ICOUNT = JWAKE                                                    REA00610
      DO 1 I = KPLUS, IIMAX                                             REA00620
      ICOUNT = ICOUNT+1                                                 REA00630
      X(I) = XWAKE(ICOUNT)                                              REA00640
```

```
      Y(I) = YWAKE(I)                                                   REA0065C
    1 XSTA(I) = XSTA(I-1)+(X(I)-X(I-1))                                 REA00660
    3 JJMAX = IIMAX-1                                                   REA00670
      WRITE (6,110) (X(I),Y(I),XSTA(I),ZERO, I=1, JJMAX)                REA00680
      WRITE (6,90) X(IIMAX),Y(IIMAX),XSTA(IIMAX),ZERO,LSTC              REA00690
      WRITE (13,120) KTE1                                               REA00700
   10 FORMAT (6(7X,I3))                                                 REA00710
   20 FORMAT (15A4,2X,2A4/I3/I5/1X,I4,I5/9X,I1)                         REA00720
   25 FORMAT (F12.6, E12.6, 4F12.6)                                     REA00730
   30 FORMAT (6F12.6)                                                   REA00740
   35 FORMAT (3F12.6,2I3)                                               REA00750
   40 FORMAT (18A4)                                                     REA00760
   50 FORMAT (8I10)                                                     REA00770
   60 FORMAT (4(7X,I3))                                                 REA00780
   70 FORMAT (2F12.6,4X,A3)                                             REA00790
   80 FORMAT (14I5)                                                     REA00800
   90 FORMAT (4F12.6,A4)                                                REA00810
  100 FORMAT (12X,F12.6)                                                REA00820
  110 FORMAT (4F12.6)                                                   RFA00830
  120 FORMAT (2X,I3)                                                    REA00840
      RETURN                                                            REA00850
      END
```

TABLE 3.1 BLINT EXEC

```
        CP MSG SUB STARTING BLINT
        CP LINK GOOD 192 192 RR ALL
        ACCESS 192 D
        CP LINK SUB 192 194 RR ALL
        ACCESS 194 C
        CP TAG DEV PRT ACCT 51841 LONGKEY EVETS  PRIOR IDLE DEST LOCAL
        &ITER = &1
        &COUNT = 1
        &LOOP -ENDLOOP &ITER
        EXEC BLAYER SUB DATA B
        &IF &RETCODE NE 0 &EXIT 1
        EXEC HESSBLI BODYCOOR DATA B
        EXEC CREPLOT1
        EXEC CREPLOT2
        &IF &RETCODE NE 0 &EXIT 2
        EXEC BLDMKER DATAMAKER DATA B
        &IF &RETCODE NE 0 &EXIT 3
        &COUNT = &COUNT + 1
        EXEC CHITER &COUNT
        &IF &RETCODE NE 0 &EXIT 4
-ENDLOOP &CONTINUE
        EXEC BLNTPLT1 &ITER
        EXEC BLNTPLT2 &ITER
```

TABLE 3.2 HESSBLI EXEC

```
        &CONTROL ALL
        FI * CLEAR
        GLOBAL TXTLIB FORTLIB
        FI 5 DISK &1 &2 &3
        TESTBAT
        &IF &RETCODE EQ 0 &GOTO -NOTBAT
        FI 06 PRINTER
        &GOTO -REST
-NOTBAT &CONTINUE
        FI 06 DISK HESSA OUTPUT A ( RECFM FA BLKSIZE 133
  -REST &CONTINUE
        FI 1 DISK FILE FT01 A
        FI 2 DISK FILE FT02 A
        FI 3 DISK FILE FT03 A
        FI 4 DISK FILE FT04 A
        FI 8 DISK FILE FT08 A
        FI 9 DISK FILE FT09 A
        FI 10 DISK FILE FT10 A
        FI 11 DISK FILE FT11 A
        FI 12 DISK FILE FT12 A
        FI 13 DISK FILE FT13 A
        FI 15 DISK FILE FT15 A
        FI 16 DISK FILE FT16 A
        FI 17 DISK HESSBLI OUTPUT E ( LRECL 80 BLKSIZE 80 RECFM FB
        FI 18 DISK HESSCP OUTPUT A (LRECL 80 BLKSIZE 80 RECFM FB
```

```
FI 19 DISK SUMDS DATA E (LRECL 80 BLKSIZE 80 RECFM FB
FI 21 DISK STREAM DATA E (RECFM FB BLKSIZE 87 LRECL 87 DISP MOD
LOAD HESSBLI  HESSA  (CLEAR NOMAP
START
&EXIT &RETCODE
```

TABLE 3.3 BLDMKER EXEC

```
FI 17 DISK HESSBLI OUTPUT E (LRECL 80 BLKSIZE 80 RECFM FB
FI 18 DISK HESSCP OUTPUT B (LRECL 80 BLKSIZE 80 RECFM FB
FI 19 DISK &1 &2 &3
LOAD BLDMKER
START
&EXIT &RETCODE
```

TABLE 3.5 BLDMKER FORTRAN

```
      DIMENSION CP(401),XW(401),X(401),F(401)                   BLD00010
  100 FORMAT (3F12.6)                                           BLD00020
  110 FORMAT (2F12.6)                                           BLD00030
  120 FORMAT (2X,I3)
```

```
      READ(19,120) IBDYPT
      IPTM1=IBDYPT-1                                                    BLD00050
   10 READ (17,100) (XW(I),X(I),F(I),I=1,IBDYPT)                        BLD00060
      CP(1)=F(1)                                                        BLD00070
      DO 20 I=2,IPTM1                                                   BLD00080
   20 CP(I) = (F(I)+F(I+1))/2.0                                         BLD00090
      DELX1=X(IBDYPT)-X(IPTM1)                                          BLD00100
      DELX2=XW(IBDYPT)-X(IBDYPT)                                        BLD00110
      DEL=DELX2/DELX1                                                   BLD00120
      CP(IBDYPT)=(1+DEL)*F(IBDYPT)-DEL*F(IPTM1)                         BLD00130
      WRITE (18,110) (XW(I),CP(I),I=1,IBDYPT)                           BLD00140
      STOP                                                              BLD00150
      END                                                               BLD00160
```

TABLE 3.6 BLAYER EXEC

```
        &CONTROL ALL
GLOBAL  TXTLIB FORTLIB VPIUTIL
        FI * CLEAR
        FI 03 DISK BODYCCUR DATA B ( LRECL 80 BLKSIZE 80 RECFM FB
        FI 18 DISK HESSCP OUTPUT B (LRECL 80 BLKSIZE 80 RECFM FB
        FI 19 DISK SUMDS DATA B (LRECL 80 BLKSIZE 80 RECFM FB
        FI 20 DISK RESTART DATA B (LRECL 80 BLKSIZE 800
        FI 05 DISK &1 &2 &3 ( LRECL 80 BLKSIZE 80 RECFM FB
        &IF &RETCODE NE 0 &EXIT &RETCODE
        TESTBAT
        &IF &RETCODE EQ 0 &GOTO -NOTBAT
        FI 06 PRINTER
        &GOTO -REST
 -NOTBAT &CONTINUE
        FI 06 DISK BLAYER OUTPUT E ( RECFM FA BLKSIZE 133
   -REST &CONTINUE
        FI 21 TERM
        FI 22 DISK VORT DATA B (LRECL 132 BLKSIZE 13200 RECFM FB
        FI 07 DISK APLNS VELDATA E (LRECL 80 BLKSIZE 80 RECFM FB
        LOAD   RESTAR READIN VELDAT ICBLINT (CLEAR NOMAP
        START
        &EXIT &RETCODE
```

TABLE 3.7   ICBLINT FORTRAN

```
C     MAIN PROGRAM                                                      MAIN  10
C                                PROGRAM ICBL                           MAIN  20
C                               NOVEMBER  1975                          MAIN  30
C                                                                       MAIN  40
C                                                                       MAIN  50
C     A PROGRAM FOR 2-D AND AXISYMMETRIC INCOMPRESSIBLE TRANSITIONAL ANDMAIN  60
C     /OR TURBULENT BOUNDARY-LAYER FLOW INCLUDING EFFECTS OF TRANSVERSE MAIN  70
C     CURVATURE                                                         MAIN  80
C                                                                       MAIN  90
C                                    BY                                 MAIN 100
C                                                                       MAIN 110
C        D.L. DWOYER, C.H. LEWIS AND E.C. ANDERSON                      MAIN 120
C        AEROSPACE AND OCEAN ENGINEERING DEPT.                          MAIN 130
C        VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY            MAIN 140
C        BLACKSBURG, VA. 24061                                          MAIN 150
C        PHONE - (703)-552-6126                                         MAIN 160
C                                                                       MAIN 170
C                                                                       MAIN 180
C     MAIN ROUTINE                                                      MAIN 190
      IMPLICIT REAL*8(A-H,O-Z)                                          MAIN 200
C                                                                       MAIN 210
C     MAIN CALLS SUBROUTINES ADDETA, CHANGE, COEF, DELTAS, DERIV3,      MAIN 220
C     GEOM, INIT, MOMENT, PRFILF, READIN, WRITE1 AND WRITE2.            MAIN 230
```

```
C                                                                       MAIN 240
C     MAIN PROVIDES CONTROL OF THE FLOW OF LOGIC IN THE PROGRAM AND     MAIN 250
C     SOME CALCULATIONS.                                                MAIN 260
C                                                                       MAIN 270
C                                                                       MAIN 280
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)MAIN 290
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),FMAIN 300
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1MAIN 310
     301),XN(102),XN2(102),Y(101),YCVDEL(101),YCVTHT(101),YY(101)       MAIN 320
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(MAIN 330
     1101),RZ(501)                                                      MAIN 340
      COMMON /CNVERG/ CONVRG,CCRNI,CRNI,DIF,DIFF,NC                     MAIN 350
      COMMON /COEFF3/ CF1,SUM                                           MAIN 360
      COMMON /COMWLL/ A1B,E1,FF,F2N1,VW                                 MAIN 370
      COMMON /CFPR/ CF                                                  MAIN 380
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO               MAIN 390
      COMMON /FRSTRM/ PFS,REINF,RHOFS,UFS                               MAIN 400
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLDMAIN 410
     1,XJ,XJFAC,Z,ZOL,RO                                                MAIN 420
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NITMAIN 430
     13,NOSE                                                            MAIN 440
      COMMON /NTEGER/ II,IPFL,IPRINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,LMAIN 450
     1AMTRB,NITTOT                                                      MAIN 460
      COMMON /NMLCRD/ ADTEST,ETAINF,XKETA                               MAIN 470
      COMMON /REF/ PREF,RHOREF,UREF                                     MAIN 480
      COMMON /VSCSTY/ EPSVD,XK1,XK2                                     MAIN 490
      COMMON /DIFP/ DIFM(301)                                           MAIN 500
      COMMON /COEFF/ CFBAR,CFBREX,CFE,CFINF,CFRES,CFREY,DEL,DELST,DELTA,MAIN 510
     1DSTARK,DSTARX,HAFCF,RETHET,REX,THET                               MAIN 520
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW     MAIN 530
      COMMON /SPTVC/ RNOSE                                              MAIN 540
      COMMON /WAK/ XKFAC                                                MAIN 550
```

```
      COMMON /TE/ KTE,KTSTTE
      COMMON /HESS/ IWRTH
      COMMON /STRTUP/ IREAD,IRSRT,KK
C
C
      DATA YES/3HYES/
C
C
      ABS(X)=DABS(X)
      SQRT(X)=DSQRT(X)
C      CALL ERRSET(208,260,2)
C
C     READ INPUT DATA; EXECUTION PARAM
C     RESTART TAPE IF ONE.
C
      CALL READIN
      IF (IREAD.EQ.1) GO TO 70
      K=0
      CALL INIT
      EPSVD=1./SQRT(REINF)
      CALL WRITE1
      CALL PRFILE
      XIOLD=XI
      XOLD=X
      Y(1)=0.0
      YY(1)=0.0
      RW=1.D0
      IF (XJFAC.GT.1.D-05.OR.XKFAC.GT.1.D-05) RW=RO
      RTVC(1)=RW
      RORW(1)=1.0D0
      IF (XKFAC.LT.1.D-05) GO TO 10
      IF (NOSE.EQ.1) RORW(1)=1.D0/DUEDS
```

```
      IF (NOSE.EQ.2) RORW(1)=RW/UE                                       MAIN 880
10    CONTINUE                                                           MAIN 890
C                                                                        MAIN 900
C     GENERATE CURVATURE CONSTANTS FOR DIFFERENT CASES; WAKE, AXISYMETRIMAIN 910
C     ECT.                                                               MAIN 920
C                                                                        MAIN 930
      RORWSQ(1)=RORW(1)**2                                               MAIN 940
      IF (NOSE.EQ.1) GO TO 30                                            MAIN 950
C                                                                        MAIN 960
C     SHARP NOSE CASE                                                    MAIN 970
C                                                                        MAIN 980
      RD=RW                                                              MAIN 990
      RDD=RW                                                             MAIN1000
      RDDD=RD                                                            MAIN1010
C                                                                        MAIN1020
C     WAKE CASE                                                          MAIN1030
C                                                                        MAIN1040
      IF (XKFAC.GT.1.D-06) RDDD=UE                                       MAIN1050
      RT=1.D0                                                            MAIN1060
      DO 20 N=2,IE                                                       MAIN1070
      Y(N)=SQRT(XI2)*XN(N)/(UE*RW)                                       MAIN1080
20    CONTINUE                                                           MAIN1090
      GO TO 50                                                           MAIN1100
30    CONTINUE                                                           MAIN1110
C                                                                        MAIN1120
C     BLUNT NOSE CASE                                                    MAIN1130
C                                                                        MAIN1140
      RD=RNOSE                                                           MAIN1150
      RDD=1.D0                                                           MAIN1160
      RDDD=RNOSE                                                         MAIN1170
C                                                                        MAIN1180
C     WAKE CASE                                                          MAIN1190
```

```
C                                                                MAIN1200
      IF (XKFAC.GT.1.D-06) RDDD=DUEDS                            MAIN1210
      RT=0.D0                                                    MAIN1220
      DO 40 N=2,IE                                               MAIN1230
      Y(N)=XN(N)*DUEDS**(XKFAC-.5D0)/DSQRT(XJFAC+XKFAC+1.D0)     MAIN1240
40    CONTINUE                                                   MAIN1250
50    CONTINUE                                                   MAIN1260
      ANG=BODANG                                                 MAIN1270
      CSORW=1.D0                                                 MAIN1280
      IF (RW.GT.1.D-06) CSORW=DCOS(ANG)/RW                       MAIN1290
C                                                                MAIN1300
C     FOR NO TRANSVERSE CURVATURE CASE                           MAIN1310
C                                                                MAIN1320
      DO 60 N=2,IE                                               MAIN1330
      RORW(N)=RORW(1)                                            MAIN1340
      RORWSQ(N)=RORWSQ(1)                                        MAIN1350
      RTVC(N)=RW                                                 MAIN1360
      YY(N)=Y(N)*EPSVD                                           MAIN1370
      IF (TVC.NE.YES) GO TO 60                                   MAIN1380
C                                                                MAIN1390
C     FOR TRANSVERSE CURVATURE CASES                             MAIN1400
C                                                                MAIN1410
      YY(N)=(-1.D0+SQRT(1.D0+2.D0*EPSVD*Y(N)*CSORW))/CSORW       MAIN1420
      RDUM=RD+RDD*YY(N)*CSORW                                    MAIN1430
      RTVC(N)=RT*RDUM                                            MAIN1440
      RORW(N)=RDUM/RDDD                                          MAIN1450
      RORWSQ(N)=RORW(N)**2                                       MAIN1460
      Y(N)=YY(N)/EPSVD                                           MAIN1470
60    CONTINUE                                                   MAIN1480
C                                                                MAIN1490
C        BEGIN MARCHING INTEGRATION                              MAIN1500
C                                                                MAIN1510
```

```
      KK=1                                                              MAIN1520
   70 DO 270 K=KK,KEND                                                  MAIN1530
      IREAD=0                                                           MAIN1540
      F1(1)=0.00                                                        MAIN1550
      F2(1)=0.00                                                        MAIN1560
      FC(1)=0.00                                                        MAIN1570
      F2N(1)=1.00                                                       MAIN1580
   80 CONTINUE                                                          MAIN1590
C                                                                       MAIN1600
C     GENERATE CURVATURE DATA FOR RESTARTED EXECUTION                   MAIN1610
C                                                                       MAIN1620
C     GENERAL 2-D CASE                                                  MAIN1630
C                                                                       MAIN1640
C     NO WAKE                                                           MAIN1650
C                                                                       MAIN1660
      RW=1.D0                                                           MAIN1670
C                                                                       MAIN1680
C     FOR WAKE CASES                                                    MAIN1690
C                                                                       MAIN1700
      IF (XJFAC.GT.1.D-05.OR.XKFAC.GT.1.D-05) RW=RO                     MAIN1710
      RTVC(1)=RW                                                        MAIN1720
      RORW(1)=1.D0                                                      MAIN1730
      RORWSQ(1)=1.D0                                                    MAIN1740
      IF (NOSE.EQ.1.AND.K.EQ.1) GO TO 110                               MAIN1750
      IF (XKFAC.LT.1.D-05) GO TO 90                                     MAIN1760
C                                                                       MAIN1770
C     SHARP NOSE, WAKE CASES                                            MAIN1780
C                                                                       MAIN1790
      RORW(1)=RW/UE                                                     MAIN1800
      RORWSQ(1)=RORW(1)**2                                              MAIN1810
   90 CONTINUE                                                          MAIN1820
      DO 100 N=2,IE                                                     MAIN1830
```

```
      Y(N)=XN(N)*PNC                                                    MAIN1840
  100 CONTINUE                                                          MAIN1850
      RDD=RW                                                            MAIN1860
      RD=RW                                                             MAIN1870
      RODD=RD                                                           MAIN1880
      IF (XKFAC.GT.1.D-06) RODD=UE                                      MAIN1890
      IF (XKFAC.GT.1.D-06.AND.RW.LT.1.D-06) RDD=1.D0                    MAIN1900
      RT=1.D0                                                           MAIN1910
      GO TO 140                                                         MAIN1920
  110 CONTINUE                                                          MAIN1930
C                                                                       MAIN1940
C      BLUNT NOSE CASE                                                  MAIN1950
C                                                                       MAIN1960
      RD=RNOSE                                                          MAIN1970
      RDD=1.D0                                                          MAIN1980
      RODD=RD                                                           MAIN1990
      IF (XKFAC.LT.1.D-06) GO TO 120                                    MAIN2000
C                                                                       MAIN2010
C     BLUNT NOSE, WAKE CASE                                             MAIN2020
C                                                                       MAIN2030
      RODD=DUEDS                                                        MAIN2040
      RORW(1)=1.D0/DUEDS                                                MAIN2050
      RORWSQ(1)=RORW(1)**2                                              MAIN2060
  120 CONTINUE                                                          MAIN2070
      RT=0.D0                                                           MAIN2080
C                                                                       MAIN2090
C     CALCULATE UNSTRETCHED COORDINATES FOR BLUNT NOSE, WAKE CASES      MAIN2100
C                                                                       MAIN2110
      DO 130 N=2,IE                                                     MAIN2120
      Y(N)=XN(N)*DUEDS**(XKFAC-.5D0)/DSQRT(XJFAC+XKFAC+1.D0)            MAIN2130
  130 CONTINUE                                                          MAIN2140
  140 CONTINUE                                                          MAIN2150
```

```
C
C     2-D CASES, CALCULATE UNSTRETCHED COORDINATES AND CURVATURE ARRAYS
C
      ANG=BODANG
      CSORW=1.D0
      IF (RW.GT.1.D-06) CSORW=DCOS(ANG)/RW
      IF (RW.LT.1.D-06.AND.XKFAC.GT.1.D-05.AND.K.NE.1) CSORW=DCOS(ANG)
      DO 150 N=2,IE
      RORW(N)=RORW(1)
      RORWSQ(N)=RORWSQ(1)
      YY(N)=Y(N)*EPSVD
      RTVC(N)=RW
C
C     GENERAL TRANSVERSE CURVATURE CASES, CALUCULATE STRETCHED AND
C     UNSTRETCHED COORDINATES AND CURVATURE ARRAYS.
      IF (TVC.NE.YES) GO TO 150
      YY(N)=(-1.D0+SQRT(1.D0+2.D0*EPSVD*Y(N)*CSORW))/CSORW
      IF (RW.LE.1.D-08.AND.K.GT.1) YY(N)=DSQRT(EPSVD*2.D0*Y(N))
      RDUM=RD+RDD*YY(N)*CSORW
      RTVC(N)=RT*RDUM
      RORW(N)=RDUM/RDDD
      RORWSQ(N)=RORW(N)**2
      Y(N)=YY(N)/EPSVD
150   CONTINUE
C
C        BEGIN ITERATION LOOP
C
160   F2N1=F2N(1)
      CALL MOMENT
      VC(1)=0.
      DO 170 N=2,IE
170   VC(N)=VC(N-1)-(XI2*(FCP(N)+FCP(N-1))+FC(N)+FC(N-1))*DN(N-1)*.5
```

```
      A1B=CRNI*F2N(1)+CCRNI*F1N(1)                                      MAIN2480
      IF (PNC.LT.1.0E-8) GO TO 190                                      MAIN2490
      CFOLD=CF                                                          MAIN2500
      CF=2.D0*UE*A1B/PNC                                                MAIN2510
      IF (RW.LT.1.D-08) CF=0.D0                                         MAIN2520
      IF (CF.GE.0.0) GO TO 180                                          MAIN2530
      WRITE (6,300) CF,UE,A1B,PNC                                       MAIN2540
      CF=CFOLD                                                          MAIN2550
180   CONTINUE                                                          MAIN2560
190   CONTINUE                                                          MAIN2570
      NIT=NIT+1                                                         MAIN2580
      NITTOT=NITTOT+1                                                   MAIN2590
      ISTOP=ISTOP+1                                                     MAIN2600
      IF (ISTOP.GT.100) GO TO 290                                       MAIN2610
      IF (K.EQ.1) GO TO 210                                             MAIN2620
      IF (NIT.LT.NIT3) GO TO 210                                        MAIN2630
      IF (DIF.LE.CONVRG) GO TO 210                                      MAIN2640
      WRITE (6,310) NIT,DX,X                                            MAIN2650
      WRITE (6,320) F2N(1),F2N1,DIF                                     MAIN2660
      IPRIIM=IPR(II-1)                                                  MAIN2670
      IF (XSTA(IPRIIM).EQ.0.0D0) GO TO 200                              MAIN2680
      IF (ABS(1.-X/XSTA(IPRIIM)).LE.1.E-6) II=II-1                      MAIN2690
200   CONTINUE                                                          MAIN2700
C                                                                       MAIN2710
C     IF TOO MANY ITERATIONS AT X, GO BACK TO PREVIOUS X AND CUT DX     MAIN2720
C     IN HALF, AND CONTINUE                                             MAIN2730
C                                                                       MAIN2740
      X=XOLD+DX/2.0                                                     MAIN2750
      DX=DX/2.0                                                         MAIN2760
      NIT=0                                                             MAIN2770
      IWRTH=0                                                           MAIN2780
      CALL DELTAS                                                       MAIN2790
```

```
      GO TO 80                                                          MAIN2800
210   CONTINUE                                                          MAIN2810
C                                                                       MAIN2820
C     CHECK FOR CONVERGENCE AND NUMBER OF ITERATIONS TO CONVERGENCE.    MAIN2830
C                                                                       MAIN2840
      IF (NIT.LT.2) GO TO 160                                           MAIN2850
      IF (DIF.GT.CONVRG) GO TO 160                                      MAIN2860
      ABST=ABS(F2(IE)-F2(IE-4))                                         MAIN2870
      IF (KADETA.EQ.0) GO TO 240                                        MAIN2880
      IF (ABST.LT.ADTEST) GO TO 220                                     MAIN2890
      SUBTST=1.0D0                                                      MAIN2900
      GO TO 230                                                         MAIN2910
220   IF (ABST.GT.ADTEST/100.0D0) GO TO 240                             MAIN2920
      SUBTST=2.0D0                                                      MAIN2930
C                                                                       MAIN2940
C     IF NEEDED, ADJUST THE ETA GRID                                    MAIN2950
C                                                                       MAIN2960
230   CALL ADDETA (SUBTST)                                              MAIN2970
      GO TO 80                                                          MAIN2980
240   CONTINUE                                                          MAIN2990
C                                                                       MAIN3000
C     CALCULATE THE BOUNDARY LAYER QUANTITIES                           MAIN3010
C                                                                       MAIN3020
      CALL COEF                                                         MAIN3030
      KRITE2=1                                                          MAIN3040
C                                                                       MAIN3050
C     WRITE OUT THE APLNS PROFILE DATA IF AT UPSTREAM BOUNDARY          MAIN3060
C     (XSTA = JSTA)                                                     MAIN3070
C                                                                       MAIN3080
      CALL VELDAT                                                       MAIN3090
      CALL WRITE2                                                       MAIN3100
C                                                                       MAIN3110
```

```
C       SAVE CURRENT VALUES AND SOLUTIONS FOR USE IN NEXT ITERATION       MAIN3120
C                                                                         MAIN3130
        XOLD=X                                                            MAIN3140
        XIOLD=XI                                                          MAIN3150
        DXIOLD=DX                                                         MAIN3160
        CALL CHANGE                                                       MAIN3170
        DO 250 N=1,IE                                                     MAIN3180
        F1(N)=F2(N)                                                       MAIN3190
        F1N(N)=F2N(N)                                                     MAIN3200
        F1NN(N)=F2NN(N)                                                   MAIN3210
250     CONTINUE                                                          MAIN3220
C                                                                         MAIN3230
C       RESET COUNTERS                                                    MAIN3240
C                                                                         MAIN3250
        NIT=0                                                             MAIN3260
        ISTOP=0                                                           MAIN3270
        IF (KSTOP.EQ.1) GO TO 260                                         MAIN3280
C                                                                         MAIN3290
C       FIND TRANSFORMED COORDINATES OF NEX                               MAIN3300
C                                                                         MAIN3310
        CALL DELTAS                                                       MAIN3320
        UERO2=UE                                                          MAIN3330
        IF (XJFAC.GT.1.D-04) UERO2=UE*RO**2                               MAIN3340
        IF (XKFAC.GT.1.D-04) UERO2=UE**3                                  MAIN3350
        CFI=2.D0*UE*A1B/PNC*EPSVD*DCOS(BODANG)                            MAIN3360
260     CONTINUE                                                          MAIN3370
        KKMX1=K+10                                                        MAIN3380
        IF (KSTOP.EQ.0) GO TO 270                                         MAIN3390
        WRITE (6,330)                                                     MAIN3400
        WRITE (6,330)                                                     MAIN3410
        GO TO 280                                                         MAIN3420
270     CONTINUE                                                          MAIN3430
```

```
      WRITE (6,340)                                                     MAIN3440
      WRITE (6,340)                                                     MAIN3450
  280 CONTINUE                                                          MAIN3460
      STOP                                                              MAIN3470
  290 WRITE (6,350)                                                     MAIN3480
      STOP                                                              MAIN3490
C                                                                       MAIN3500
C                                                                       MAIN3510
C                                                                       MAIN3520
  300 FORMAT (1H0,16HCF, UE, A1B, PNC/4E15.6///)                        MAIN3530
  310 FORMAT (1H0,///6H  NIT=,I5,5H  DX=,E14.6,4H  X=,E14.6/)           MAIN3540
  320 FORMAT (9H  F2N(1)=,1PE14.6,7H  F2N1=,E14.6,6H  DIF=,E14.6///)    MAIN3550
  330 FORMAT (1H1,/////50X,9H THE END ,14H X=XSTA(IIMAX))               MAIN3560
  340 FORMAT (1H1,/////50X,9H THE END ,7H K=KEND)                       MAIN3570
  350 FORMAT (25H  STOP *** ISTOP.GT.100//)                             MAIN3580
      END                                                               MAIN3590
```

```
      SUBROUTINE ADDETA (SUBTST)                                        ADDO  10
      IMPLICIT REAL*8(A-H,O-Z)                                          ADDO  20
C                                                                       ADDO  30
C     SUBROUTINE ADDETA CALLS SUBROUTINES INTER3 AND DERIV3.            ADDO  40
C                                                                       ADDO  50
C     SUBROUTINE ADDETA IS CALLED BY MAIN.                              ADDO  60
C                                                                       ADDO  70
C     SUBROUTINE ADDETA INCREASES ETAINF IF F(IE) AND F(IE-4) DIFFER BY ADDO  80
C     MORE THAN ADTEST.                                                 ADDO  90
C     GENERATES NEW ETA SPACING AND INTERPOLATES FOR NEW VELOCITY       ADDO 100
C     PROFILE.                                                          ADDO 110
C                                                                       ADDO 120
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)ADDO 130
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),FADDO 140
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1ADDO 150
     301),XN(102),XN2(102),Y(101),YCVDEL(101),YOVTHT(101),YY(101)       ADDO 160
      COMMON /COMWLL/ A1B,E1,FF,F2N1,VW                                 ADDO 170
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLDADDO 180
     1,XJ,XJFAC,Z,ZOL,RO                                                ADDO 190
      COMMON /INTGR/ IF,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NITADDO 200
     13,NOSE                                                            ADDO 210
      COMMON /NTEGER/ II,IPFL,IPRINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,LADDO 220
     1AMTRB,NITTOT                                                      ADDO 230
      COMMON /NMLCRD/ ADTEST,ETAINF,XKETA                               ADDO 240
C                                                                       ADDO 250
C                                                                       ADDO 260
      FLOAT(M)=DFLOAT(M)                                                ADDO 270
C                                                                       ADDO 280
C     INCREASE OR DECREASE ETAINF, DEPENDING ON TEST-SUBTST             ADDO 290
C                                                                       ADDO 300
      IF (SUBTST.EQ.2.0D0) GO TO 10                                     ADDO 310
      WRITE (6,80) ISTOP,K                                              ADDO 320
```

```
      ETAIN2=ETAINF+.1D0*ETAINF                                         ADDO 330
      GO TO 20                                                          ADDO 340
10    WRITE (6,100) ISTOP,K                                             ADDO 350
      ETAIN2=ETAINF-0.1D0*ETAINF                                        ADDO 360
C                                                                       ADDO 370
C     CALCULATE NEW ETA PROFILE                                         ADDO 380
C                                                                       ADDO 390
20    CONTINUE                                                          ADDO 400
      IF (XKETA.EQ.1.0) DETA1=ETAIN2/FLOAT(IE-1)                        ADDO 410
      IF (XKETA.NE.1.0) DETA1=ETAIN2*(XKETA-1.0)/(XKETA**IM-1.0)        ADDO 420
      DN2(1)=DETA1                                                      ADDO 430
      XN2(1)=0.0                                                        ADDO 440
      DO 30 N=1,IE                                                      ADDO 450
      DN2(N+1)=DN2(N)*XKETA                                             ADDO 460
      XN2(N+1)=XN2(N)+DN2(N)                                            ADDO 470
30    CONTINUE                                                          ADDO 480
      DO 60 N=1,IE                                                      ADDO 490
      IF (XN2(N).GE.ETAINF) GO TO 50                                    ADDO 500
      JC=0                                                              ADDO 510
40    JC=JC+1                                                           ADDO 520
      IF (XN2(N).GT.XN(JC)) GO TO 40                                    ADDO 530
      IF (JC.LT.2) JC=2                                                 ADDO 540
      IF (JC.GT.(IE-1)) JC=IE-1                                         ADDO 550
      CALL INTER3 (XN2(N),XN(JC-1),XN(JC),XN(JC+1),F1(JC-1),F1(JC),F1(JCADDO 560
     1+1),F2(N))                                                        ADDO 570
      GO TO 60                                                          ADDO 580
50    F2(N)=1.0                                                         ADDO 590
60    CONTINUE                                                          ADDO 600
C                                                                       ADDO 610
C     INTERPOLATE OLD SOLUTION TO NEW ETA GRID FOR 1ST GUESS ON CURRENT ADDO 620
C     SOLUTION                                                          ADDO 630
C                                                                       ADDO 640
```

```
      CALL DERIV3 (F2,XN2,IE,1,F2N)                                     ADDO 650
      CALL DERIV3 (F2N,XN2,IE,1,F2NN)                                   ADDO 660
      VC(1)=VW                                                          ADDO 670
C                                                                       ADDO 680
C     PUT NEW GRIDS AND PROFILES INTO WORKING ARRAYS                    ADDO 690
C                                                                       ADDO 700
      DO 70 N=1,IE                                                      ADDO 710
      DN(N)=DN2(N)                                                      ADDO 720
      XN(N)=XN2(N)                                                      ADDO 730
      F1(N)=F2(N)                                                       ADDO 740
      F1N(N)=F2N(N)                                                     ADDO 750
      F1NN(N)=F2NN(N)                                                   ADDO 760
      FC(N)=F1(N)                                                       ADDO 770
      FCN(N)=F1N(N)                                                     ADDO 780
      FCP(N)=0.0                                                        ADDO 790
      IF (N.EQ.1) GO TO 70                                              ADDO 800
      VC(N)=VC(N-1)-(F2(N)+F2(N-1))*DN(N-1)*0.50                        ADDO 810
70    CONTINUE                                                          ADDO 820
      ETAINF=ETAIN2                                                     ADDO 830
      WRITE (6,90) DX,XKETA,ETAINF,ADTEST,X                             ADDO 840
      RETURN                                                            ADDO 850
C                                                                       ADDO 860
C                                                                       ADDO 870
C                                                                       ADDO 880
80    FORMAT (1H0,9X,42HINTERMEDIATE PROFILE DATA-ETAINF INCREASED,2X,6HADDO 890
     1ISTOP=,I3,3H K=,I3)                                               ADDO 900
90    FORMAT (1H0,2X,3HDX=,F12.6,1X,6HXKETA=,F12.6,1X,7HETAINF=,F12.6,1XADDO 910
     1,7HADTEST=,F12.6,1X,2HX=,F12.6)                                   ADDO 920
100   FORMAT (1H0,9X,42HINTERMEDIATE PROFILE DATA-ETAINF DECREASED,2X,6HADDO 930
     1ISTOP=,I3,3H K=,I3)                                               ADDO 940
      END                                                               ADDO 950
```

```
      SUBROUTINE BLUNT1
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE BLUNT1 CALLS SUBROUTINE FD5.
C
C     SUBROUTINE BLUNT1 IS CALLED BY SUBROUTINE GEOM.
C
C     SUBROUTINE BLUNT1 CALCULATES THE EDGE AND REFERENCE PROPERTIES
C     FOR THE STAGNATION POINT OF A BLUNT BODY.
C
      COMMON /ARRAY2/ PZ(501),UFZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(
     1101),RZ(501)
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO
      COMMON /FRSTRM/ PFS,REINF,RHOFS,UFS
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLD
     1,XJ,XJFAC,Z,ZOL,RO
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
      COMMON /REF/ PREF,RHOREF,UREF
      COMMON /STAG/ P10,PSTAG
      COMMON /VSCSTY/ EPSVD,XK1,XK2
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW
      COMMON /WAK/ XKFAC
C
C
      ALOG10(X)=DLOG10(X)
      SQRT(X)=DSQRT(X)
C
C
      P10=PSTAG/(RHOFS*UFS*UFS)
      DO 10 J=1,IIMAX
      PZ(J)=PZ(J)*P10
```

```
      UEZ(J)=SQRT(2.*(P10-PZ(J)))                                       BLAO 330
10    CONTINUE                                                          BLAO 340
      BODANG=DARCOS(-1.0D0)/2.0D0                                       BLAO 350
      Z=ZA(1)                                                           BLAO 360
      R0=0.0                                                            BLAO 370
      PE=P10                                                            BLAO 380
      UE=0.0                                                            BLAO 390
      PP=0.0                                                            BLAO 400
C                                                                       BLAO 410
C     FIND STAGNATION POINT EDGE VELOCITY GRADIENT BY INTERPOLATION AND BLAO 420
C     REFLECTION AROUND THE ORIGIN                                      BLAO 430
C                                                                       BLAO 440
      CALL FD5 (X,-XSTA(3),-XSTA(2),XSTA(1),XSTA(2),XSTA(3),-UEZ(3),-UEZBLAO 450
     1(2),UEZ(1),UEZ(2),UEZ(3),DUEDS)                                   BLAO 460
      UER02=0.0                                                         BLAO 470
      PNC=SQRT((XKFAC+1.D0)/((XJFAC+1.)*DUEDS))                         BLAO 480
      BETA=0.5D0                                                        BLAO 490
      PES0=PE                                                           BLAO 500
      WRITE (6,20) P10                                                  BLAO 510
      WRITE (6,30) UE,PE                                                BLAO 520
      RETURN                                                            BLAO 530
C                                                                       BLAO 540
C                                                                       BLAO 550
20    FORMAT (1H ,2X,4HP10=,D15.7)                                      BLAO 560
30    FORMAT (9H   UES0= ,D15.7,9H   PES0= ,D15.7)                      BLAO 570
      END                                                               BLAO 580
```

```
      SUBROUTINE BLUNT2 (IFLAG)                                         BLBO  10
      IMPLICIT REAL*8(A-H,O-Z)                                          BLBO  20
C                                                                       BLBO  30
C     SUBROUTINE BLUNT2 CALLS SUBROUTINES FD5, INTER5 AND ZRO           BLBO  40
C                                                                       BLBO  50
C     SUBROUTINE BLUNT2 IS CALLED BY SUBROUTINE EGPROP.                 BLBO  60
C                                                                       BLBO  70
C     SUBROUTINE BLUNT2 CALCULATES THE EDGE PROPERTIES FOR A BLUNT BODY BLBO  80
C     AT THE VALUES OF X AFTER THE INITIAL VALUE OF X.                  BLBO  90
C                                                                       BLBO 100
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(BLBO 110
     1101),RZ(501)                                                      BLBO 120
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO               BLBO 130
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLDBLBO 140
     1,XJ,XJFAC,Z,ZOL,RO                                                BLBO 150
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NITBLBO 160
     13,NOSE                                                            BLBO 170
      COMMON /STAG/ P10,PSTAG                                           BLBO 180
      COMMON /VSCSTY/ EPSVD,XK1,XK2                                     BLBO 190
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW     BLBO 200
      COMMON /DRAG/ CDV1,CDV2,XCP,XCP1,CP(501),CP1(501)                 BLBO 210
      COMMON /HESS/ IWTH,ICARD2,ICARD3,ICARD4,NN,IBDN,HEDR(10),CASE,KHESBLBO 220
     1S                                                                 BLBO 230
C                                                                       BLBO 240
C                                                                       BLBO 250
      SQRT(X)=DSQRT(X)                                                  BLBO 260
C                                                                       BLBO 270
C                                                                       BLBO 280
      IF (X.LT.XSTA(3)) GO TO 30                                        BLBO 290
      J=0                                                               BLBO 300
10    J=J+1                                                             BLBO 310
      IF (X.GT.XSTA(J)) GO TO 10                                        BLBO 320
```

```
      IF (J.LT.3) J=3
      IF (J.GT.(IIMAX-2)) J=IIMAX-2
C
C     FIND PROPERTIES IF PAST THE THIRD STATICN
C
      CALL ZRO (IIMAX,X,Z,RO)
      IF (RO.LT.0.D0) RO=0.D0
      CALL FD5 (Z,ZA(J-2),ZA(J-1),ZA(J),ZA(J+1),ZA(J+2),RZ(J-2),RZ(J-1),
     1RZ(J),RZ(J+1),RZ(J+2),TANBOD)
      BODANG=DATAN(TANBOD)
      CALL INTER5 (X,XSTA(J-2),XSTA(J-1),XSTA(J),XSTA(J+1),XSTA(J+2),PZ(
     1J-2),PZ(J-1),PZ(J),PZ(J+1),PZ(J+2),PE)
      UE=SQRT(2.*(P10-PE))
      CALL FD5 (X,XSTA(J-2),XSTA(J-1),XSTA(J),XSTA(J+1),XSTA(J+2),UEZ(J-
     12),UEZ(J-1),UEZ(J),UEZ(J+1),UEZ(J+2),DUEDS)
      IF (IFLAG.EQ.0) GC TO 20
      IF (KHESS.EQ.0) GC TO 20
      CALL INTER5 (X,XSTA(J-2),XSTA(J-1),XSTA(J),XSTA(J+1),XSTA(J+2),CP(
     1J-2),CP(J-1),CP(J),CP(J+1),CP(J+2),XCP)
      CALL INTER5 (X,XSTA(J-2),XSTA(J-1),XSTA(J),XSTA(J+1),XSTA(J+2),CP1
     1(J-2),CP1(J-1),CP1(J),CP1(J+1),CP1(J+2),XCP1)
   20 CONTINUE
      PP=-UE*DUEDS
      RETURN
   30 CONTINUE
      CALL ZRO (IIMAX,X,Z,RO)
      CALL FD5 (Z,ZA(1),ZA(2),ZA(3),ZA(4),ZA(5),RZ(1),RZ(2),RZ(3),RZ(4),
     1RZ(5),TANBOD)
      BODANG=DATAN(TANBOD)
      CALL INTER5 (X,-XSTA(3),-XSTA(2),XSTA(1),XSTA(2),XSTA(3),PZ(3),PZ(
     12),PZ(1),PZ(2),PZ(3),PE)
      CALL FD5 (X,-XSTA(3),-XSTA(2),XSTA(1),XSTA(2),XSTA(3),-UEZ(3),-UEZ
```

```
1(2),UEZ(1),UEZ(2),UEZ(3),DUEDS)                                       BLBO 650
 IF (IFLAG.EQ.0) GC TO 40                                              BLBO 660
 IF (KHESS.FQ.0) GC TO 40                                              RLBO 670
 CALL INTERS (X,-XSTA(3),-XSTA(2),XSTA(1),XSTA(2),XSTA(3),CP(3),CP(BLBO 680
12),CP(1),CP(2),CP(3),XCP)                                             BLBO 690
 CALL INTERS (X,-XSTA(3),-XSTA(2),XSTA(1),XSTA(2),XSTA(3),CP1(3),CPBLBO 700
11(2),CP1(1),CP1(2),CP1(3),XCP1)                                       BLPO 710
40    UE=SQRT(2.*(P10-PE))                                             BLBO 720
 GO TC 20                                                              BLBO 730
 END                                                                   BLBO 740
```

```
      SUBROUTINE CHANGE                                                 CHAO
      IMPLICIT REAL*8(A-H,O-Z)                                          CHAO
C                                                                       CHAO
C     SUBROUTINE CHANGE IS CALLED BY MAIN.                              CHAO
C                                                                       CHAO
C     SUBROUTINE CHANGE ADJUSTS THE X STEP SIZE IF NECESSARY TO OBTAIN  CHAO
C     A SOLUTION AT A SPECIFIED VALUE OF X.                             CHAO
C     WHEN INSTANTANEOUS TRANSITICN IS TO BE CALCULATED, THIS SUBROUTINECHAO
C     CHANGES THE SOLUTION PROCEDURE FROM LAMINAR TO TURBULENT AND      CHAO
C     RESETS DELTA X.                                                   CHAO
C     THIS SUBROUTINE DOUBLES/HALVES THE X STEP SIZE IF NIT IS LESS THANCHAO
C     NIT2 OR GREATER THAN NIT1 RESPECTIVELY. DX CAN NEVER BE SET       CHAO
C     GREATER THAN DXMAX.                                               CHAO
C                                                                       CHAO
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)CHAO
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),FCHAO
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1CHAO
     301),XN(102),XN2(102),Y(101),YOVDEL(101),YOVTHT(101),YY(101)       CHAO
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(CHAO
     1101),RZ(501)                                                      CHAO
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLDCHAO
     1,XJ,XJFAC,Z,ZOL,RO                                                CHAO
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NITCHAO
     13,NOSE                                                            CHAO
      COMMON /NTEGER/ II,IPFL,IPRINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,LCHAO
     1AMTRB,NITTOT                                                      CHAO
      COMMON /TRANS/ ATR,CHICRT,CHIMAX,GAMMA,XBAR,XIBAR,KTRANS,KONSET   CHAO
      COMMON /TE/ KTE,KTSTTE                                            CHAO
      COMMON /HESS/ IWRTH                                               CHAO
C                                                                       CHAO
C                                                                       CHAO
      ABS(X)=DABS(X)                                                    CHAO
```

```
C
C
      IF (X.EQ.(XSTA(1)+1.D-04)) DX=DX1
      IF (X.GT.(XSTA(1)+1.D-04)) GO TO 10
      X=X+DX
      GO TO 80
10    CONTINUE
      KONSET=IPR(IPRINT)
      IF (NIT.GE.0) GO TO 20
      DX=DX/2.D0
      X=X-DX
      GO TO 80
C     ADJUST DX USING ITER CNTR, NIT
20    IF (NIT.GT.NIT1) GO TO 30
      DX=2.D0*DX
      IF (DX.GT.DXMAX) DX=DXMAX
      GO TO 50
30    IF (NIT.LT.NIT2) GO TO 50
      DX=.5D0*DX
C
C     CHECK IF DX IS TOO SMALL, IF SO, HALT EXECUTION, SOLUTION WILL NO
C     CONVERGE.  USUALLY MEANS FLOW HAS SEPARATED.
C
      IF (DX.GT.DX1/100.D0) GO TO 40
      WRITE (6,110)
      STOP
40    CONTINUE
C
C     IF START OF TRANSITION, SET DX FOR GAMMA
C
50    IF (X.EQ.XSTA(KONSET).AND.KONSET.NE.IIMAX) DX=XSTA(KONSET)*6.32121
     1D-02
```

```
C
C     ADJUST DX TO SOLVE AT IPR STATIONS
C
      DO 70 II=1,IPRINT
      IF (XSTA(IPR(II)).LE.X) GO TO 70
      IF (XSTA(IPR(II)).GT.(X+1.25D0*DX)) GO TO 60
C     SET FINAL VALUE
      DX=XSTA(IPR(II))-X
      X=XSTA(IPR(II))
      IWRTH=1
      GO TO 80
60    X=X+DX
      GO TO 80
70    CONTINUE
C     LAST STATION SOLVED, TERMINATE RUN
      KSTOP=1
      RETURN
80    DXOLD=DX
      XOLD=X
C
C     CHECK IF INSTANTANEOUS TRANSITION
C
      IF (KTRNSN.EQ.IIMAX) GO TO 90
      IF (X.LT.XSTA(KTRNSN)) GO TO 90
      GAMMA=1.D0
      LAMTRB=2
90    CONTINUE
      IF (KTE.EQ.IIMAX) GO TO 100
      IF (X.LE.XSTA(KTE)) GO TO 100
      KTSTTE=1
100   CONTINUE
      RETURN
```

```
C                                                              CHAO 970
110   FORMAT (1H0,24HQUITTING DUE TO SMALL DX)                 CHAO 980
      END                                                      CHAO 990
```

```
      SUBROUTINE COEF                                                   COEO  10
      IMPLICIT REAL*8(A-H,O-Z)                                          COEO  20
C                                                                       COEO  30
C     SUBROUTINE COEF CALLS SUBROUTINE INTERP.                          COEO  40
C                                                                       COEO  50
C     SUBROUTINE COEF IS CALLED BY MAIN.                                COEO  60
C                                                                       COEO  70
C     SUBROUTINE COEF COMPUTES FLOW COEFFICIENTS SUCH AS SKIN FRICTION  COEO  80
C     COEFFICIENT, ETC.                                                 COEO  90
C     ALSO COMPUTES DELTA, THETA, AND DELTA*.                           COEO 100
C                                                                       COEO 110
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)COEO 120
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),FCOEO 130
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1COEO 140
     301),XN(102),XN2(102),Y(101),YOVDEL(101),YOVTHT(101),YY(101)       COEO 150
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(COEO 160
     1101),RZ(501)                                                      COEO 170
      COMMON /COEFF/ CFBAR,CFBREX,CFE,CFINF,CFRES,CFREY,DEL,DELST,DELTA,COEO 180
     1DSTARK,DSTARX,HAFCF,RETHET,REX,THET                               COEO 190
      COMMON /COEFF2/ DELCRF,DELOX,DSAXOR,DSTCOL,DSTORF,DSTOTH,DSTOX,RORCOEO 200
     1EFL,THODEL,THOREF,XOREFL,ZOREFL                                   COEO 210
      COMMON /COEFF3/ CF1,SUM,SUM1,SUMT                                 COEO 220
      COMMON /COMWLL/ A1B,E1,FF,F2N1,VW                                 COEO 230
      COMMON /CFPR/ CF                                                  COEO 240
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO               COEO 250
      COMMON /FRSTRM/ PFS,REINF,RHOFS,UFS                               COEO 260
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLDCOEO 270
     1,XJ,XJFAC,Z,ZOL,RO                                                COEO 280
      COMMON /INTGR/ IE,IIMAX,IM,IPPNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NITCOEO 290
     13,NOSE                                                            COEO 300
      COMMON /NTEGER/ II,IPFL,IPRINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,LCOEO 310
     1AMTRB,NITTOT                                                      COEO 320
```

```
      COMMON /TRANS/ ATR,CHICRT,CHIMAX,GAMMA,XBAR,XIBAR,KTRANS,KONSET    CDEO 330
      COMMON /VSCSTY/ EPSVD,XK1,XK2                                      CDEO 340
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW      CDEO 350
      COMMON /LWALL/ UOUPLS(101),YPLUS(101),UDEF(101),UPLUS              CDEO 360
      COMMON /TE/ KTE,KTSTTE                                             CDEO 370
      COMMON /DRAG/ CDV1,CDV2,XCP,XCP1,CP(501),CP1(501),ROMAX            CDEO 380
      COMMON /STAG/ P10,PSTAG,RHO                                        CDEO 390
      COMMON /WKPRL/ RSTAR                                               CDEO 400
      COMMON /HESS/ IWTH,ICARD2,ICARD3,ICARD4,NN,IBDN,HEDR(10),CASE,KHESCDEO 410
     1S                                                                  CDEO 420
      DATA YES/3HYES/                                                    CDEO 430
      DIMENSION FD(101)                                                  CDEO 440
C                                                                        CDEO 450
C                                                                        CDEO 460
      SQRT(X)=DSQRT(X)                                                   CDEO 470
C                                                                        CDEO 480
      IF (X.EQ.XSTA(1)) GO TO 60                                         CDEO 490
C                                                                        CDEO 500
      PI=DARCOS(-1.D0)                                                   CDEO 510
C                                                                        CDEO 520
C                                                                        CDEO 530
      RESE=UE*X                                                          CDEO 540
      REX=REINF*RESE                                                     CDEO 550
      SQREX=SQRT(REX)                                                    CDEO 560
      CFRFS=CF*SQREX                                                     CDEO 570
      CFE=CF*EPSVD/(UE*UE)                                               CDEO 580
      CFINF=CF*EPSVD                                                     CDEO 590
      CFRFY=CFRFS*EPSVD                                                  CDEO 600
      IF (X.EQ.0.0) GO TO 20                                             CDEO 610
      CALL INTERP (0.995D0,FC,Y,IE,DELTA)                                CDEO 620
      DO 10 I=1,IE                                                       CDEO 630
      YOVDEL(I)=Y(I)/DELTA                                               CDEO 640
```

```
10    CONTINUE
20    CONTINUE
C
C        AVG. CF*SQRT(REX)
C
      IF (K.EQ.1) GO TO 30
      IF (KTSTTE.GT.0) GO TO 40
      CFBREX=2.*2.**.5*A1B
      CF2=CFINF*DCOS(BODANG)
C
C     CALCULATE INCREMENTAL SKIN DRAG DUE TO DX
C
      DSUM=0.50*DX*(CF2+CF1)
C
C      RUNNING SUM OF SKIN DRAG
C
      SUM=SUM+DSUM
      CFBREX=SUM/X*SQREX
      IF (KHESS.EQ.0) GO TO 30
      REFA=PI*ROMAX**2
      C=4.D0*PI/UFS**2/REFA/RHO
      CDV2=C*(XCP-XCP1)*RO*DSIN(BODANG)
C
C     CALCULATE INCREMENTAL PRESSURE DRAG
C
      DSUM1=0.5*DX*(CDV1+CDV2)
C
C     RUNNING SUM OF PRESSURE DRAG
C
      SUM1=SUM1+DSUM1
      CDV1=CDV2
      SUMT=SUM+SUM1
```

```
30    CONTINUE                                                          COEO 970
      HAFCF=CF*EPSVD/2.0                                                COEO 980
      IF (X.EQ.0.0) GO TO 40                                            COEO 990
      CFBAR=CFBREX/SQREX                                                COEO1000
40    CONTINUE                                                          COEO1010
C                                                                       COEO1020
C     CALCULATE MOMENTUM THICKNESS                                      COEO1030
C                                                                       COEO1040
      THET=0.0                                                          COEO1050
      DO 50 I=2,IE                                                      COEO1060
      FAC1=FC(I)*(1.0D0-FC(I))                                          COEO1070
      FAC2=FC(I-1)*(1.0D0-FC(I-1))                                      COEO1080
      THET=THET+0.5D0*(FAC1+FAC2)*DN(I-1)*PNC*EPSVD                     COEO1090
50    CONTINUE                                                          COEO1100
      FASANG=BODANG                                                     COEO1110
C                                                                       COEO1120
C     THETA FOR TRAVSVERSE CURVATURE CASE FOR A 0.0 OR FINITE WALL RADIUCOEO1130
C                                                                       COEO1140
      IF (TVC.EQ.YES.AND.RW.GT.1.D-08) THET=RW/DCOS(FASANG)*(-1.0D0+DSQRCOEO1150
     1T(1.0D0+2.0D0*DCOS(FASANG)*THET/RW))                              COEO1160
      IF (TVC.EQ.YES.AND.RW.LE.1.D-08) THET=DSQRT(2.D0*THET)            COEO1170
60    DELST=0.0                                                         COEO1180
C                                                                       COEO1190
C      INTEGRATE TO FIND THE DISPLACEMENT THICKNESS                     COEO1200
C                                                                       COEO1210
      DO 70 I=2,IE                                                      COEO1220
      FAC1=1.D0-FC(I)                                                   COEO1230
      FAC2=1.D0-FC(I-1)                                                 COEO1240
      DELST=DELST+0.5D0*(FAC1+FAC2)*DN(I-1)*PNC*EPSVD                   COEO1250
70    CONTINUE                                                          COEO1260
      IF (TVC.EQ.YES.AND.RW.GT.1.D-08) DELST=RW/DCOS(FASANG)*(-1.0D0+DSQCOEO1270
     1RT(1.0D0+2.0D0*DCOS(FASANG)*DELST/RW))                            COEO1280
```

```
      IF (TVC.EQ.YES.AND.RW.LE.1.D-08) DELST=DSQRT(2.D0*DELST)          COE01290
      IF (X.EQ.XSTA(1)) RETURN                                          COE01300
C                                                                       COE01310
C       CALCULATE PRINTED BOUNDARY LAYER QUANTITIES                     COE01320
C                                                                       COE01330
      RETHET=UE*THET/(EPSVD*EPSVD)                                      COE01340
      DEL=DELTA*EPSVD                                                   COE01350
      ROREFL=RO/REFLEN                                                  COE01360
      XOREFL=X/REFLEN                                                   COE01370
      THOREF=THET/REFLEN                                                COE01380
      DSTORF=DELST/REFLEN                                               COE01390
      DSTOX=DELST/X                                                     COE01400
      DELOX=DEL/X                                                       COE01410
      DELORF=DEL/REFLEN                                                 COE01420
      IF (XJFAC.NE.0.0) DSTRAX=RO*(SQRT(1.0+2.0*DELST/RO)-1.0)          COE01430
      IF (TVC.EQ.YES) DSTRAX=DELST                                      COE01440
      IF (XJFAC.EQ.0.0) DSTRAX=0.0                                      COE01450
      ZOREFL=Z/REFLEN                                                   COE01460
      THODEL=THET/DEL                                                   COE01470
      DSTODL=DELST/DEL                                                  COE01480
      DSTOTH=DELST/THET                                                 COE01490
      DSAXOR=DSTRAX/REFLEN                                              COE01500
      DO 80 N=1,IE                                                      COE01510
80    YOVTHT(N)=YY(N)/THET                                              COE01520
      IF (KTSTTE.GT.0) GO TO 100                                        COE01530
      UPLUS=SQRT(UE*EPSVD*DABS(A1B)/PNC)                                COE01540
      DO 90 I=1,IE                                                      COE01550
      UOUPLS(I)=FC(I)/UPLUS*UE                                          COE01560
      YPLUS(I)=YY(I)*UPLUS/EPSVD**2                                     COE01570
      UDEF(I)=UE*(FC(I)-1.D0)/UPLUS                                     COE01580
90    CONTINUE                                                          COE01590
100   CONTINUE                                                          COE01600
```

```
      VCRT=DSQRT(2.D0*XI)/UE/DELTA                                  COE01610
      THETA=BODANG                                                  COE01620
      RDEL=RO+DELTA*DCOS(THETA)                                     COE01630
      ZDEL=Z-DELTA*DSIN(THETA)                                      COE01640
      WRITE (22,120) VCRT,Z,RO,ZDEL,RDEL,DELTA,THETA                COE01650
      IF (KTSTTE.LT.1) RETURN                                       COE01660
      BIGUD=1.D0-FC(1)                                              COE01670
      DO 110 I=1,IE                                                 COE01680
      FD(I)=1.D0-FC(I)                                              COE01690
  110 FD(I)=FD(I)/BIGUD                                             COE01700
      CALL INTERP (0.5D0,FD,YY,IE,RSTAR)                            COE01710
      RETURN                                                        COE01720
C                                                                   COE01730
  120 FORMAT (7E15.6)                                               COE01740
      END                                                           COE01750
```

```
      SUBROUTINE CONE1
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE CONE1 CALLS SUBROUTINES FD5, INTER5 AND ZRO.
C
C     SUBROUTINE CONE1 IS CALLED BY SUBROUTINE GEOM.
C
C     SUBROUTINE CONE1 CALCULATES THE EDGE AND REFERENCE PROPERTIES FOR
C     THE FIRST CALCULATED POINT ON A SHARP NOSE BODY.
C
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(
     1101),RZ(501)
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO
      COMMON /FRSTPM/ PFS,REINF,RHOFS,UFS
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLD
     1,XJ,XJFAC,Z,ZOL,FO
      COMMON /REF/ PREF,RHOREF,UREF
      COMMON /STAG/ P10,PSTAG
      COMMON /VSCSTY/ EPSVD,XK1,XK2
      COMMON /STAG4/ UECN,PECN
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW
C
C
      SIN(X)=DSIN(X)
      SQRT(X)=DSQRT(X)
C
      P10=PSTAG/(RHOFS*UFS*UFS)
      DO 10 J=1,IIMAX
      PZ(J)=PZ(J)*P10
      UEZ(J)=DSQRT(2.D0*(P10-PZ(J)))
```

```
10     CONTINUE                                                          CDAO 330
       CALL ZRO (IIMAX,X,Z,RO)                                           CDAO 340
       IF (RO.LT.0.DO) RO=0.DO                                           CDAO 350
       CALL FD5 (Z,ZA(1),ZA(2),ZA(3),ZA(4),ZA(5),RZ(1),RZ(2),RZ(3),RZ(4),CDAO 360
      1RZ(5),TANBOD)                                                     CDAO 370
       BODANG=DATAN(TANBOD)                                              CDAO 380
       CALL INTER5 (X,XSTA(1),XSTA(2),XSTA(3),XSTA(4),XSTA(5),PZ(1),PZ(2)CDAO 390
      1,PZ(3),PZ(4),PZ(5),PE)                                            CDAO 400
       CALL FD5 (X,XSTA(1),XSTA(2),XSTA(3),XSTA(4),XSTA(5),UEZ(1),UEZ(2),CDAO 410
      1UEZ(3),UEZ(4),UEZ(5),DUEDS)                                       CDAO 420
       UE=SQRT(2.*(P10-PE))                                              CDAO 430
       RETURN                                                            CDAO 440
       END                                                               CDAO 450
```

```
      SUBROUTINE CONE2 (IFLAG)
      IMPLICIT REAL*8(A-H,O-Z)
C     G
C     SUBROUTINE CONE2 CALLS SUBROUTINES FD5, INTER5 AND ZRO.
C
C     SUBROUTINE CONE2 IS CALLED BY SUBROUTINE EGPROP.
C
C     SUBROUTINE CONE2 CALCULATES THE EDGE PROPERTIES FOR A SHARP NOSE
C     BODY AT THE VALUES OF X AFTER THE INITIAL VALUE OF X.
C
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(
     1101),RZ(501)
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
      COMMON /STAG/ P10,PSTAG
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLD
     1,XJ,XJFAC,Z,ZOL,RO
      COMMON /STAG4/ UECN,PECN
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW
C
      COMMON /HESS/ IWRTH,ICARD2,ICARD3,ICARD4,NN,IBDN,HEDR(10),CASE,KHE
     1SS
      COMMON /DRAG/ CDV1,CDV2,XCP,XCP1,CP(501),CP1(501)
C
      COS(X)=DCOS(X)
      SQRT(X)=DSQRT(X)
C
C
      IF (X.LT.XSTA(3)) GO TO 30
      J=0
10    J=J+1
```

```
      IF (X.GT.XSTA(J)) GC TO 10                                        COBO 330
      IF (J.LT.3) J=3                                                   CCBO 340
      IF (J.GT.(IIMAX-2)) J=IIMAX-2                                     COBO 350
      CALL ZRO (IIMAX,X,Z,RO)                                           COBO 360
      IF (RO.LT.0.DO) RO=0.DO                                           CORO 370
      CALL FD5 (Z,ZA(J-2),ZA(J-1),ZA(J),ZA(J+1),ZA(J+2),RZ(J-2),RZ(J-1),CCBO 380
     1RZ(J),RZ(J+1),RZ(J+2),TANBOD)                                     CCBO 390
      BCDANG=DATAN(TANBOD)                                              CUBO 400
      CALL INTER5 (X,XSTA(J-2),XSTA(J-1),XSTA(J),XSTA(J+1),XSTA(J+2),PZ(CCBO 410
     1J-2),PZ(J-1),PZ(J),PZ(J+1),PZ(J+2),PE)                            COBO 420
      UE=SQRT(2.*(PIO-PE))                                              COBO 430
      CALL FD5 (X,XSTA(J-2),XSTA(J-1),XSTA(J),XSTA(J+1),XSTA(J+2),UEZ(J-COBO 440
     12),UEZ(J-1),UEZ(J),UEZ(J+1),UEZ(J+2),DUEDS)                       CUBO 450
      IF (IFLAG.EQ.0) GO TO 20                                          CCBO 460
      IF (KHESS.EQ.0) GC TO 20                                          COBO 470
      CALL INTER5 (X,XSTA(J-2),XSTA(J-1),XSTA(J),XSTA(J+1),XSTA(J+2),CP(CCBO 480
     1J-2),CP(J-1),CP(J),CP(J+1),CP(J+2),XCP)                           COBO 490
      CALL INTER5 (X,XSTA(J-2),XSTA(J-1),XSTA(J),XSTA(J+1),XSTA(J+2),CP1CUBO 500
     1(J-2),CP1(J-1),CP1(J),CP1(J+1),CP1(J+2),XCP1)                     CCBO 510
   20 CONTINUE                                                          CCBO 520
      PP=-UE*DUEDS                                                      CCBO 530
      RETURN                                                            CCBO 540
   30 CONTINUE                                                          COBO 550
      CALL ZRO (IIMAX,X,Z,RO)                                           COBO 560
      CALL FD5 (Z,ZA(1),ZA(2),ZA(3),ZA(4),ZA(5),RZ(1),RZ(2),RZ(3),RZ(4),COBO 570
     1RZ(5),TANBOD)                                                     CCBO 580
      BCDANG=DATAN(TANBCD)                                              COBO 590
      CALL INTER5 (X,XSTA(1),XSTA(2),XSTA(3),XSTA(4),XSTA(5),PZ(1),PZ(2)CCRO 600
     1,PZ(3),PZ(4),PZ(5),PE)                                            CCBU 610
      CALL FD5 (X,XSTA(1),XSTA(2),XSTA(3),XSTA(4),XSTA(5),UEZ(1),UEZ(2),COBO 620
     1UEZ(3),UEZ(4),UEZ(5),DUEDS)                                       COBO 630
      IF (IFLAG.EQ.0) GO TO 40                                          COBO 640
```

```
      IF (KHESS.EQ.0) GO TO 40                                          COBO 650
      CALL INTER5 (X,-XSTA(3),-XSTA(2),XSTA(1),XSTA(2),XSTA(3),CP(3),CP(COBO 660
     12),CP(1),CP(2),CP(3),XCP)                                         COBO 670
      CALL INTER5 (X,-XSTA(3),-XSTA(2),XSTA(1),XSTA(2),XSTA(3),CP1(3),CPCOBO 680
     11(2),CP1(1),CP1(2),CP1(3),XCP1)                                   COBO 690
   40 UE=SQRT(2.*(P10-PE))                                              COBO 700
      GO TO 20                                                          COBO 710
      END                                                               COBO 720
```

```
      SUBROUTINE DELTAS                                                 DELO  10
      IMPLICIT REAL*8(A-H,O-Z)                                          DELO  20
C                                                                       DELO  30
C     SUBROUTINE DELTAS CALLS SUBROUTINE EGPROP.                        DELO  40
C                                                                       DELO  50
C     SUBROUTINE DELTAS IS CALLED BY MAIN AND SUBROUTINE INIT.          DELO  60
C                                                                       DELO  70
C     SUBROUTINE DELTAS CALCULATES DS - DELTA XI, THE TRANSFORMATION    DELO  80
C     FACTOR - PNC, AND THE PRESSURE GRADIENT TERM - BETA.              DELO  90
C     INTEGRATION BY SIMPSONS RULE IS USED FOR CALCULATING DS.          DELO 100
C                                                                       DELO 110
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO               DELO 120
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLDDELO 130
     1,XJ,XJFAC,Z,ZOL,RO                                                DELO 140
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NITDELO 150
     13,NCSE                                                            DELO 160
      COMMON /WAK/ XKFAC                                                DELO 170
      COMMON /INTACT/ BETAS                                             DELO 180
      COMMON /HESS/ IWRTH,ICARD2,ICARD3,ICARD4,NN,IBDN,HEDR(10),CASE,KHEDELO 190
     1SS                                                                DELO 200
C                                                                       DELO 210
C                                                                       DELO 220
      SQRT(X)=DSQRT(X)                                                  DELO 230
C                                                                       DELO 240
C                                                                       DELO 250
      HA=X-DX*0.50                                                      DELO 260
      X1=X                                                              DELO 270
      X=HA                                                              DELO 280
      CALL EGPROP (0)                                                   DELO 290
      X=X1                                                              DELO 300
      CALL EGPROP (1)                                                   DELO 310
C                                                                       DELO 320
```

```
C        AXISYMETRIC, WAKE CASES                                        DELO 330
C                                                                       DELO 340
      RP=RO                                                             DELO 350
      UEHA=UE                                                           DELO 360
      ROHA=RO                                                           DELO 370
      RPP=RP                                                            DELO 380
      IF (XJFAC.LT.0.0010) GO TO 10                                     DELO 390
      GO TO 20                                                          DELO 400
C                                                                       DELO 410
C     2-D, NO WAKE CASES                                                DELO 420
C                                                                       DELO 430
10    CONTINUE                                                          DELO 440
      RP=1.00                                                           DELO 450
      RPP=RP                                                            DELO 460
      ROHA=1.00                                                         DELO 470
20    CONTINUE                                                          DELO 480
      IF (XKFAC.LT.0.0010) GO TO 30                                     DELO 490
C                                                                       DELO 500
C     2-D OR AXISYMETRIC, WITH WAKE                                     DELO 510
C                                                                       DELO 520
      RP=UE                                                             DELO 530
      RPP=RO                                                            DELO 540
      ROHA=UEHA                                                         DELO 550
30    CONTINUE                                                          DELO 560
      DS=(UERO2+4.*UEHA*ROHA*ROHA+UE*RP*RP)*CX/6.                       DELO 570
      XI=XIOLD+DS                                                       DELO 580
      XI2=2.00*XI                                                       DELO 590
      PNC=SQRT(XI2)/UE                                                  DELO 600
      IF (RPP.GT.1.D-08) PNC=PNC/RPP                                    DELO 610
      BETA=XI2*DUEDS/((UE*RP)**2)                                       DELO 620
      IF (KHESS.EQ.0) RETURN                                            DELO 630
      IF (BETA.LT.BETAS) BETA=BETAS                                     DELO 640
```

```
      RETURN                                                            DELO 650
      END                                                               DELO 660
```

```
      SUBROUTINE DERIV3 (F,X,IMAX,IMIN,FP)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE DERIV3 CALLS SUBROUTINE FD3.
C
C     SUBROUTINE DERIV3 IS CALLED BY MAIN AND SUBROUTINES ADDETA, EFFMU
C     AND SOLVE.
C
C      SUBROUTINE DERIV3 CALCULATES THE FIRST DERIVATIVES OF F WITH
C     RESPECT TO X AND RETURNS THE ARRAY FP.
C
      DIMENSION F(101), X(102), FP(101)
C
C
      DO 10 J=IMIN,IMAX
      K=J
      IF (K.LT.(IMIN+1)) K=IMIN+1
      IF (K.GT.(IMAX-1)) K=IMAX-1
      CALL FD3 (X(J),X(K-1),X(K),X(K+1),F(K-1),F(K),F(K+1),FP(J))
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EFFMU (LAMTRB,II,DLSTRK)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE EFFMU CALLS SUBROUTINES DERIV3 AND INTERP.
C
C     SUBROUTINE EFFMU IS CALLED BY SUBROUTINE MOMENT.
C
C     CALCULATES THE TERMS AOB AND AOBP WHICH, FOR THE TURBULENT
C     CASE, INCLUDE THE EDDY VISCOSITY - EPSPL.
C     FOR THE TRANSITION REGIME, THE EDDY VISCOSITY IS REDUCED BY THE
C     TRANSITION INTERMITTENCY FACTOR - GAMMA.
C
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),F
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1
     301),XN(102),XN2(102),Y(101),YCVDEL(101),YOVTHT(101),YY(101)
      COMMON /ARRAY2/ PZ(501),UFZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(
     1101),RZ(501)
      COMMON /CFPR/ CF
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLD
     1,XJ,XJFAC,Z,ZOL,RO
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
      COMMON /TRANS/ ATR,CHICRT,CHIMAX,GAMMA,XBAR,XIBAR,KTRANS,KONSET
      COMMON /VSCSTY/ EPSVO,XK1,XK2
      COMMON /COEFF/ CFBAR,CFBREX,CFE,CFINF,CFRES,CFREY,DEL,DELST,DELTA,
     1DSTARK,DSTARX,HAFCF,RETHET,REX,THET
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RCRW(101),RORWSQ(101),RW
      COMMON /TE/ KTE,KTSTTE
      COMMON /WAK/ XKFAC
      DATA YES/3HYES/
```

```
C                                                                       EFFO 330
C                                                                       EFFO 340
      ABS(X)=DABS(X)                                                    EFFO 350
      EXP(X)=DEXP(X)                                                    EFFO 360
      SQRT(X)=DSQRT(X)                                                  EFFO 370
      TANH(X)=DTANH(X)                                                  EFFO 380
      DSTARK=0.                                                         EFFO 390
C                                                                       EFFO 400
C                                                                       EFFO 410
C        LAMTRB = 1 LAMINAR B.L. ,=2 TURBULENT B.L.                     EFFO 420
C                                                                       EFFO 430
C                                                                       EFFO 440
      IF (LAMTRB.EQ.2) GO TO 30                                         EFFO 450
C                                                                       EFFO 460
C        LAMINAR                                                        EFFO 470
C                                                                       EFFO 480
C                                                                       EFFO 490
C        COEFFICIENTS FOR MOMENTUM EQUATION                             EFFO 500
C                                                                       EFFO 510
      DO 10 I=1,IE                                                      EFFO 520
      AOB(I)=RORWSQ(I)                                                  EFFO 530
10    CONTINUE                                                          EFFO 540
      CALL DERIV3 (AOB,XN,IE,1,AOBP)                                    EFFO 550
      IF ((X-XSTA(1)).LT.1.E-7) RETURN                                  EFFO 560
      IF (PNC.LT.1.0E-10) RETURN                                        EFFO 570
      CHIMAX=0.                                                         EFFO 580
C                                                                       EFFO 590
C        CALCULATES VORTICITY REYNOLDS NUMBER                           EFFO 600
C                                                                       EFFO 610
      DO 20 N=2,IE                                                      EFFO 620
      CHI(N)=0.0                                                        EFFO 630
      IF (CHI(N).GT.CHIMAX) CHIMAX=CHI(N)                               EFFO 640
```

```
20    CONTINUE                                                          EFFO 650
      IF (KTRANS.EQ.0) RETURN                                           EFFO 660
C                                                                       EFFO 670
C        INITIATES TRANSITION                                           EFFO 680
C                                                                       EFFO 690
      LAMTRB=2                                                          EFFO 700
      DX=DX/10.                                                         EFFO 710
      DXOLD=DXOLD/10.                                                   EFFO 720
      WRITE (6,190)                                                     EFFO 730
      WRITE (6,200)                                                     EFFO 740
C     XZERO=X                                                           EFFO 750
      XZERO=X-XSTA(1)                                                   EFFO 760
      XLAMDA=(XBAR-1.)*XZERO/4.                                         EFFO 770
C                                                                       EFFO 780
C     WHEN USING THIS TRANSITION MODEL,THE VALUE OF XBAR SHOULD BE      EFFO 790
C     SIXTY-FIVE PERCENT GREATER WHEN THE REICHARDT EDDY VISCOSITY LAW  EFFO 800
C     IS USED THAN THE VALUE OF XBAR USED WITH THE VAN DRIEST EDDY      EFFO 810
C     VISCOSITY LAW.  THE USER IS ADVISED THAT THE TRANSITION MODEL IS  EFFO 820
C     TENTATIVE ONLY AND THAT DUE CARE MUST BE EXERCISED IN INTERPRETINGEFFO 830
C     THE RESULTS OBTAINED WITH THIS TRANSITION MODEL.                  EFFO 840
C                                                                       EFFO 850
30    CONTINUE                                                          EFFO 860
      IF (KTRANS.EQ.0) GO TO 40                                         EFFO 870
      XIBAR=((X-XSTA(1))-XZERO)/XLAMDA                                  EFFO 880
      GAMMA=1.-EXP(-ATR*XIBAR**2)                                       EFFO 890
      IF (KTRANS.EQ.2) GO TO 40                                         EFFO 900
      IF (KVSLAW.EQ.0.AND.XIBAR.LT.2.0) GO TO 40                        EFFO 910
      IF (KVSLAW.NE.0.AND.XIBAR.LT.4.0) GO TO 40                        EFFO 920
C                                                                       EFFO 930
C        KTRANS IS SET EQUAL TO 2 AT THE END OF THE TRANSITION REGIME   EFFO 940
C                                                                       EFFO 950
      KTRANS=2                                                          EFFO 960
```

```
      WRITE (6,210)                                                     EFF0 970
      WRITE (6,200)                                                     EFF0 980
   40 CONTINUE                                                          EFF0 990
C                                                                       EFF01000
C        TURBULENT                                                      EFF01010
C                                                                       EFF01020
      FAC1=0.0                                                          EFF01030
      IF (KVSLAW.NE.0) GO TO 70                                         EFF01040
      DO 60 I=1,IM                                                      EFF01050
C                                                                       EFF01060
C        INNER EDDY VISCOSITY LAW, REICHARDT EQUATION                   EFF01070
C                                                                       EFF01080
      FAC=CF/(2.*EPSVD)                                                 EFF01090
      IF (FAC.GT.0.0) GO TO 50                                          EFF01100
      WRITE (6,180) FAC,CF,EPSVD                                        EFF01110
      FAC=1.0E-6                                                        EFF01120
   50 CONTINUE                                                          EFF01130
      FAC1=SQRT(FAC)                                                    EFF01140
      EPSPL(I)=(.4*Y(I)*FAC1-4.4*TANH(Y(I)*FAC1/11.))*GAMMA             EFF01150
   60 CONTINUE                                                          EFF01160
      GO TO 100                                                         EFF01170
   70 CONTINUE                                                          EFF01180
C                                                                       EFF01190
C        INNER EDDY VISCOSITY LAW, VAN DRIEST EQUATION                  EFF01200
C                                                                       EFF01210
      FAC=SQRT(.5D0*CF)                                                 EFF01220
      IF (FAC.GT.0.0) GO TO 80                                          EFF01230
      WRITE (6,180) FAC,CF,EPSVD                                        EFF01240
      FAC=1.0E-6                                                        EFF01250
   80 CONTINUE                                                          EFF01260
      ROF=1.D0                                                          EFF01270
      DO 90 I=1,IM                                                      EFF01280
```

```
      IF (XJFAC.GT.1.D-04.OR.XKFAC.GT.1.D-04) ROF=RTVC(I)               EFF01290
      XDOUG1=(XK1*UE*Y(I))**2*ROF/(EPSVD*XI2**.5)                       EFF01300
      XDOUG2=Y(I)/(26.*EPSVD**.5)                                       EFF01310
      XDOUG3=0.D0                                                       EFF01320
      IF ((XDOUG2*FAC).LT.70) XDOUG3=EXP(-XDOUG2*FAC)                   EFF01330
      EPSPL(I)=XDOUG1*((1.-XDOUG3)**2)*ABS(FCN(I))*GAMMA                EFF01340
90    CONTINUE                                                          EFF01350
100   CONTINUE                                                          EFF01360
      DSTARK=0.0                                                        EFF01370
      DO 110 I=2,IE                                                     EFF01380
      FAC1=1.D0-FC(I)                                                   EFF01390
      FAC2=1.D0-FC(I-1)                                                 EFF01400
      DSTARK=DSTARK+0.5D0*DN(I-1)*(FAC1+FAC2)*PNC*EPSVD                 EFF01410
110   CONTINUE                                                          EFF01420
C                                                                       EFF01430
      ANG=BODANG                                                        EFF01440
      IF (TVC.EQ.YES.AND.RW.GT.1.D-08) DSTARK=RW/DCOS(ANG)*(-1.D0+DSQRT(EFF01450
     11.D0+2.D0*DCOS(ANG)*DSTARK/RW))                                   EFF01460
      IF (TVC.EQ.YES.AND.RW.LE.1.D-08) DSTARK=DSQRT(2.D0*DSTARK)        EFF01470
C                                                                       EFF01480
C         INTERPOLATE FOR DELTA WITH INTERP AND TLU                     EFF01490
C                                                                       EFF01500
      CALL INTERP (0.9950D0,FC,Y,IE,DELTA)                              EFF01510
C                                                                       EFF01520
C        OUTER EDDY VISCOSITY LAW                                       EFF01530
C                                                                       EFF01540
      CHIMAX=0.                                                         EFF01550
      DO 120 I=2,IE                                                     EFF01560
      CHI(I)=0.0                                                        EFF01570
      IF (CHI(I).GT.CHIMAX) CHIMAX=CHI(I)                               EFF01580
      GAMF=1.0/(1.0+5.50*(Y(I)/DELTA)**6)                               EFF01590
      EPSO(I)=XK2*UE*DSTARK*GAMF*GAMMA/EPSVD**2                         EFF01600
```

```
120    CONTINUE                                                    EFF01610
       II=1                                                        EFF01620
       IF (KTSTTE.EQ.1) GO TO 140                                  EFF01630
       IF (XK2.GT.100.0) GO TO 160                                 EFF01640
       DO 130 I=1,IE                                               EFF01650
       IF (EPSPL(I).LT.EPSO(I)) GO TO 130                          EFF01660
       II=I                                                        EFF01670
       GO TO 140                                                   EFF01680
130    CONTINUE                                                    EFF01690
       II=IE                                                       EFF01700
140    DO 150 I=II,IE                                              EFF01710
150    EPSPL(I)=EPSO(I)                                            EFF01720
160    CONTINUE                                                    EFF01730
       DO 170 I=1,IE                                               EFF01740
       AOB(I)=(1.+EPSPL(I))*RORWSQ(I)                              EFF01750
170    CONTINUE                                                    EFF01760
       CALL DERIV3 (AOB,XN,IE,1,AOBP)                              EFF01770
       RETURN                                                      EFF01780
C                                                                  EFF01790
C                                                                  EFF01800
C                                                                  EFF01810
180    FORMAT (1H ,14H FAC,CF,EPSVD /4F15.6///)                    EFF01820
190    FORMAT (28H1           TRANSITION BEGINS/)                  EFF01830
200    FORMAT (1H1)                                                EFF01840
210    FORMAT (26H1           TRANSITION ENDS/)                    EFF01850
       END                                                         EFF01860
```

```
      SUBROUTINE EGPROP (IFLAG)                                         EGPO  10
      IMPLICIT REAL*8(A-H,O-Z)                                          EGPO  20
C                                                                       EGPO  30
C     SUBROUTINE EGPROP CALLS SUBROUTINES BLUNT2 AND CONE2.             EGPO  40
C                                                                       EGPO  50
C     SUBROUTINE EGPROP IS CALLED BY SUBROUTINE DELTAS.                 EGPO  60
C                                                                       EGPO  70
C     SUBROUTINE EGPROP OBTAINS EDGE PROPERTIES FOR SPECIFIED GEOMETRY  EGPO  80
C     FOR VALUES OF X AFTER THE INITIAL VALUE OF X.                     EGPO  90
C                                                                       EGPO 100
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NITEGPO 110
     13,NOSE                                                            EGPO 120
C                                                                       EGPO 130
C                                                                       EGPO 140
      GO TO (10,20), NOSE                                               EGPO 150
10    CONTINUE                                                          EGPO 160
      CALL BLUNT2 (IFLAG)                                               EGPO 170
      RETURN                                                            EGPO 180
20    CONTINUE                                                          EGPO 190
      CALL CONE2 (IFLAG)                                                EGPO 200
      RETURN                                                            EGPO 210
      END                                                               EGPO 220
```

```
      SUBROUTINE FD3 (X,X1,X2,X3,F1,F2,F3,FX)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE FD3 IS CALLED BY SUBROUTINE DERIV3.
C
C     SUBROUTINE FD3 CALCULATES THE FIRST DERIVATIVE-FX-CORRESPONDING
C     TO POINT X USING 3 POINT LAGRANGIAN DIFFERENTIATION FORMULA.
C
C     ASSUMES X1 .LE. X .LE. X3.
C
      A1=2.0*X-X2-X3
      A2=2.0*X-X1-X3
      A3=2.0*X-X1-X2
      D1=(X1-X2)*(X1-X3)
      D2=(X2-X1)*(X2-X3)
      D3=(X3-X1)*(X3-X2)
      C1=A1/D1
      C2=A2/D2
      C3=A3/D3
      FX=C1*F1+C2*F2+C3*F3
      RETURN
      END
```

```
      SUBROUTINE FD5 (X,X1,X2,X3,X4,X5,F1,F2,F3,F4,F5,FX)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE FD5 IS CALLED BY SUBROUTINES BLUNT1, BLUNT2, CONE1
C     AND CONE2.
C
C     SUBROUTINE FD5 CALCULATES THE FIRST DERIVATIVE-FX-CORRESPONDING
C     TO POINT X USING 5 POINT LAGRANGIAN DIFFERENTIATION FORMULA
C
C     ASSUMES X1 .LE. X .LE. X5.
C
      A1=(X-X4)*(X-X5)*(2.0*X-X2-X3)+(X-X2)*(X-X3)*(2.0*X-X4-X5)
      A2=(X-X4)*(X-X5)*(2.0*X-X1-X3)+(X-X1)*(X-X3)*(2.0*X-X4-X5)
      A3=(X-X4)*(X-X5)*(2.0*X-X1-X2)+(X-X1)*(X-X2)*(2.0*X-X4-X5)
      A4=(X-X3)*(X-X5)*(2.0*X-X1-X2)+(X-X1)*(X-X2)*(2.0*X-X3-X5)
      A5=(X-X3)*(X-X4)*(2.0*X-X1-X2)+(X-X1)*(X-X2)*(2.0*X-X3-X4)
      D1=(X1-X2)*(X1-X3)*(X1-X4)*(X1-X5)
      D2=(X2-X1)*(X2-X3)*(X2-X4)*(X2-X5)
      D3=(X3-X1)*(X3-X2)*(X3-X4)*(X3-X5)
      D4=(X4-X1)*(X4-X2)*(X4-X3)*(X4-X5)
      D5=(X5-X1)*(X5-X2)*(X5-X3)*(X5-X4)
      C1=A1/D1
      C2=A2/D2
      C3=A3/D3
      C4=A4/D4
      C5=A5/D5
      FX=C1*F1+C2*F2+C3*F3+C4*F4+C5*F5
      RETURN
      END
```

```
      SUBROUTINE GEOM
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE GEOM CALLS SUBROUTINES BLUNT1 AND CONE1.
C
C     SUBROUTINE GEOM IS CALLED BY MAIN AND SUBROUTINE INIT.
C
C     SUBROUTINE GEOM OBTAINS EDGE PROPERTIES FOR SPECIFIED GEOMETRY
C     FOR THE INITIAL VALUE OF X.
C
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
C
C     NOSE=1 BLUNT BODY
C     NOSE=2 SHARP NOSE BODY
C
      GO TO (10,20), NOSE
10    CALL BLUNT1
      RETURN
20    CONTINUE
      CALL CONE1
      RETURN
      END
```

```
      SUBROUTINE INIT                                                   INIO  10
      IMPLICIT REAL*8(A-H,O-Z)                                          INIO  20
C                                                                       INIO  30
C     SUBROUTINE INIT CALLS SUBROUTINES DELTAS AND GEOM.                INIO  40
C                                                                       INIO  50
C     SUBROUTINE INIT IS CALLED BY MAIN.                                INIO  60
C                                                                       INIO  70
C     THIS SUBROUTINE PROVIDES INITIALIZATION OF DATA FOR THE PROGRAM.  INIO  80
C                                                                       INIO  90
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(INIO 100
     1101),RZ(501)                                                      INIO 110
      COMMON /CNVERG/ CONVRG,CCRNI,CRNI,DIF,DIFF,NC                     INIO 120
      COMMON /COEFF3/ CF1,SUM,SUM1,SUMT                                 INIO 130
      COMMON /COMWLL/ A16,E1,FF,F2N1,VW                                 INIO 140
      COMMON /CFPR/ CF                                                  INIO 150
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO               INIO 160
      COMMON /FRSTRM/ PFS,REINF,RHOFS,UFS                               INIO 170
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLDINIO 180
     1,XJ,XJFAC,Z,ZOL,RO                                                INIO 190
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NITINIO 200
     13,NOSE                                                            INIO 210
      COMMON /NTEGER/ II,IPFL,IPRINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,LINIO 220
     1AMTRB,NITTOT                                                      INIO 230
      COMMON /REF/ PREF,RHOREF,UREF                                     INIO 240
      COMMON /STAG/ P10,PSTAG                                           INIO 250
      COMMON /TRANS/ ATR,CHICRT,CHIMAX,GAMMA,XBAR,XIBAR,KTRANS          INIO 260
      COMMON /VSCSTY/ EPSVD,XK1,XK2                                     INIO 270
      COMMON /STAG4/ UECN,PECN                                          INIO 280
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW     INIO 290
      COMMON /TE/ KTE,KTSTTE                                            INIO 300
      COMMON /WAK/ XKFAC                                                INIO 310
      COMMON /DRAG/ CDV1,CDV2,XCP,XCP1                                  INIO 320
```

```
      DATA YES,NO/3HYES,2HNO/
C
C
      BODANG=0.DO
      IF (TVC.EQ.NO) RW=1.DO
      XSTA(IIMAX+1)=-2.E6
C      XSTA = -2.E6 TERMINATES SOLUTION
      KEP=0
      XJAY=XJFAC
      CF=1.00
      VW=0.0
      KSTUP=0
      II=2
      IPRNT=1
      SUM=0.0
      CDV1=0.0
      SUM1=0.0
      SUMT=0.0
      Z=ZA(1)
      X=XSTA(1)
      XI=0.00
      XI2=2.00*XI
      XIOLD=XI
      XOLD=X
      DXOLD=0.0
      DS=1.00
      CCRNI=1.00-CRNI
      E1=0.00
      FIN=1.00
      FF=0.00
      NIT=0
      NITTOT=0
```

```
      ISTOP=0                                                           INIO 650
      CHIMAX=0.0                                                        INIO 660
      GAMMA=1.                                                          INIO 670
      IF (LAMTRB.EQ.1) GAMMA=0.0                                        INIO 680
      XIBAR=0.                                                          INIO 690
      KTSTTE=0                                                          INIO 700
      IM=IE-1                                                           INIO 710
      JJ=1                                                              INIO 720
      RHOFS=2.*(PSTAG-PFS)/(UFS*UFS)                                    INIO 730
      CALL GEOM                                                         INIO 740
      DX1=DX                                                            INIO 750
      IF (NOSE.EQ.2) GO TO 10                                           INIO 760
      IF (NOSE.EQ.1) GO TO 20                                           INIO 770
      IF (LAMTRB.NE.2) GO TO 20                                         INIO 780
10    CONTINUE                                                          INIO 790
      X=XSTA(1)+1.D-02                                                  INIO 800
      IF (TVC.EQ.YES) X=XSTA(1)+0.01D0                                  INIO 810
      IF (TVC.EQ.YES) X=XSTA(1)+0.00025                                 INIO 820
      RO=X*DSIN(BODANG)                                                 INIO 830
      DX=0.010                                                          INIO 840
      IF (TVC.EQ.YES) DX=.00025                                         INIO 850
      XOLD=X                                                            INIO 860
      UERO2=UE                                                          INIO 870
      IF (TVC.EQ.YES) UERO2=UE*RO**2                                    INIO 880
      IF (XKFAC.GT.1.D-04) UERO2=UE**3                                  INIO 890
      CALL DELTAS                                                       INIO 900
      DX=DX1                                                            INIO 910
20    CONTINUE                                                          INIO 920
      RETURN                                                            INIO 930
C                                                                       INIO 940
      END                                                               INIO 950
```

```
      SUBROUTINE INTERP (XX,XN,F2,IE,FF)                                INPO  10
      IMPLICIT REAL*8(A-H,O-Z)                                          INPO  20
C                                                                       INPO  30
C     SUBROUTINE INTERP CALLS FUNCTION TLU.                             INPO  40
C                                                                       INPO  50
C     SUBROUTINE INTERP IS CALLED BY SUBROUTINES COEF AND EFFMU.        INPO  60
C                                                                       INPO  70
C     SUBROUTINE INTERP USES FUNCTION TLU TO INTERPOLATE IN ARRAY F2    INPO  80
C     FOR THE VALUE FF CORRESPONDING TO THE VALUE XX IN ARRAY XN.       INPO  90
C     IF XX .LT. XN(1) .OR. XX .GT. XN(IE), FF IS SET EQUAL TO F2(IE)   INPO 100
C     AND A MESSAGE IS PRINTED.                                         INPO 110
C                                                                       INPO 120
      DIMENSION XN(101), F2(101)                                        INPO 130
C                                                                       INPO 140
C                                                                       INPO 150
      FF=TLU(IE,F2,XN,XX,NFLAG)                                         INPO 160
      IF (NFLAG.NE.1) RETURN                                            INPO 170
      WRITE (6,10)                                                      INPO 180
      FF=F2(IE)                                                         INPO 190
      RETURN                                                            INPO 200
C                                                                       INPO 210
C                                                                       INPO 220
C                                                                       INPO 230
10    FORMAT (1HO,10X,38HINADAQUATE TABLE FOR SUBROUTINE INTERP,//11X,43INPO 240
     1HSTANDARD FIXUP TAKEN - EXECUTION CONTINUING)                     INPO 250
      END                                                               INPO 260
```

```
      SUBROUTINE INTER3 (X,X1,X2,X3,F1,F2,F3,F)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE INTER3 IS CALLED BY SUBROUTINE ADDETA.
C
C     SUBROUTINE INTER3 INTERPOLATES FOR THE VALUE F CORRESPONDING TO
C     POINT X USING 3 POINT LAGRANGIAN INTERPOLATION.
C
C     ASSUMES X1 .LE. X .LE. X3.
C
      A1=(X-X2)*(X-X3)
      A2=(X-X1)*(X-X3)
      A3=(X-X1)*(X-X2)
      D1=(X1-X2)*(X1-X3)
      D2=(X2-X1)*(X2-X3)
      D3=(X3-X1)*(X3-X2)
      C1=A1/D1
      C2=A2/D2
      C3=A3/D3
      F=C1*F1+C2*F2+C3*F3
      RETURN
      END
```

```
      SUBROUTINE INTER5 (X,X1,X2,X3,X4,X5,F1,F2,F3,F4,F5,F)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE INTER5 IS CALLED BY SUBROUTINES BLUNT1, BLUNT2, CONE1,
C     CONE2, INIT, ZRO, AND INTRP5.
C
C     SUBROUTINE INTER5 INTERPOLATES FOR THE VALUE F CORRESPONDING TO
C     POINT X USING 5 POINT LAGRANGIAN INTERPOLATION FORMULA.
C
C     ASSUMES X1 .LE. X .LE. X5.
C
      A1=(X-X2)*(X-X3)*(X-X4)*(X-X5)
      A2=(X-X1)*(X-X3)*(X-X4)*(X-X5)
      A3=(X-X1)*(X-X2)*(X-X4)*(X-X5)
      A4=(X-X1)*(X-X2)*(X-X3)*(X-X5)
      A5=(X-X1)*(X-X2)*(X-X3)*(X-X4)
      D1=(X1-X2)*(X1-X3)*(X1-X4)*(X1-X5)
      D2=(X2-X1)*(X2-X3)*(X2-X4)*(X2-X5)
      D3=(X3-X1)*(X3-X2)*(X3-X4)*(X3-X5)
      D4=(X4-X1)*(X4-X2)*(X4-X3)*(X4-X5)
      D5=(X5-X1)*(X5-X2)*(X5-X3)*(X5-X4)
      C1=A1/D1
      C2=A2/D2
      C3=A3/D3
      C4=A4/D4
      C5=A5/D5
      F=C1*F1+C2*F2+C3*F3+C4*F4+C5*F5
      RETURN
      END
```

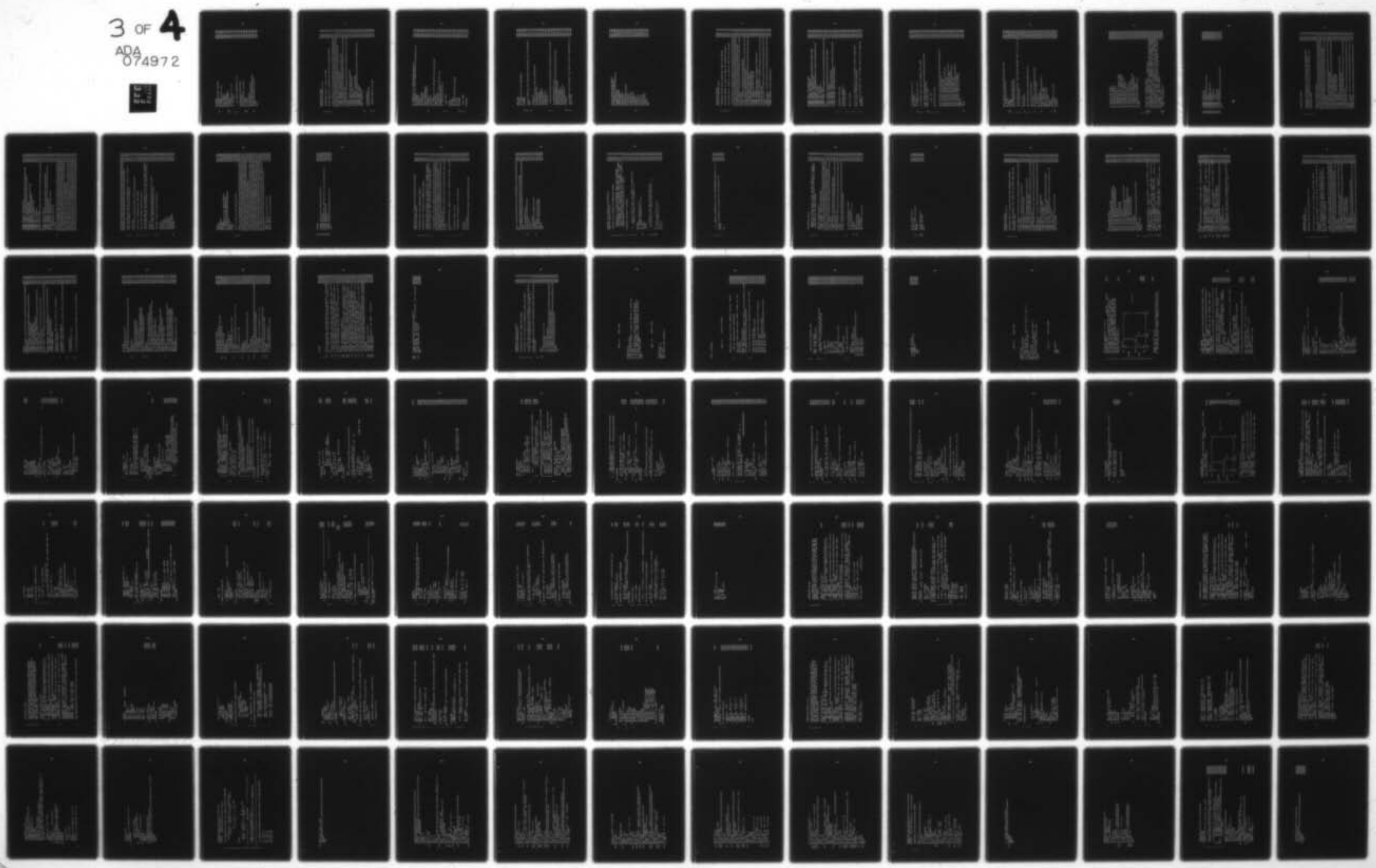
```
      SUBROUTINE INTRP3 (XX,X,Y,NPNTS,YY)
      IMPLICIT REAL*8(A-H,O-Z)
C
C         SUBROUTINE INTRP3 SETS UP THE CALLING ARGUMENT FOR
C         SUBROUTINE INTER3
C
C         YY IS THE VALUE RETURNED FROM ARRAY Y
C         WHICH CORRESPONDS TO THE VALUE XX IN ARRAY X
C
      DIMENSION X(1), Y(1)
C
      JC=0
10    JC=JC+1
      IF (XX.GT.X(JC)) GO TO 10
      IF (JC.LT.2) JC=2
      IF (JC.GT.(NPNTS-1)) JC=NPNTS-1
      CALL INTER3 (XX,X(JC-1),X(JC),X(JC+1),Y(JC-1),Y(JC),Y(JC+1),YY)
      RETURN
      END
```

```
      SUBROUTINE MOMENT
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE MOMENT CALLS SUBROUTINES EFFMU AND SOLVE.
C
C     SUBROUTINE MOMENT IS CALLED BY MAIN.
C
C     THIS SUBROUTINE PROVIDES THE SOLUTION FOR THE MOMENTUM EQUATION.
C
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),F
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1
     301),XN(102),XN2(102),Y(101),YOVDEL(101),YOVTHT(101),YY(101)
      COMMON /CNVERG/ CONVRG,CCRNI,CRNI,DIF,DIFF,NC
      COMMON /COEFF/ CFBAR,CFPREX,CFE,CFINF,CFRES,CFREY,DEL,DELST,DELTA,
     1DSTARK,DSTARX,HAFCF,RETHET,REX,THET
      COMMON /COMWLL/ A1B,E1,FF,F2N1,VW
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLD
     1,XJ,XJFAC,Z,ZCL,RO
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
      COMMON /NTEGER/ II,IPFL,IPRINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,L
     1AMTRB,NITTOT
      COMMON /DIFP/ DIFM(301)
      COMMON /TE/ KTE,KTSTTE
C
C
      ABS(X)=DABS(X)
C
C
      DIF=0.D0
```

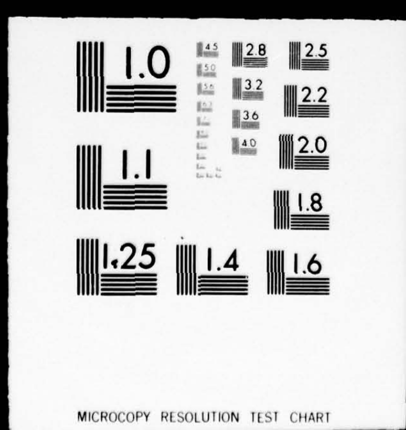
```
      CALL EFFMU (LAMTRB,KEP,DSTARK)
C
C        MOMENTUM EQUATION SOLUTION
C        MOMENTUM EQUATION COEFFICIENTS
C
      DO 10 N=2,IM
      A1(N)=(AOBP(N)-VC(N))/AOB(N)
      A2(N)=-BETA*FC(N)/AOB(N)
      A3(N)=BETA/AOB(N)
      A4(N)=-2.*XI*FC(N)/AOB(N)
10    CONTINUE
      BC1=E1
      BC2=FF
      IF (KTSTTE.EQ.0) GO TO 20
      BC1=1.D0
      BC2=0.D0
20    CONTINUE
      CALL SOLVE (F1NN,F1N,F1,F2NN,F2N,F2,BC1,BC2,CRNI)
      KF2=0
      DO 30 N=2,IE
      IF (F2(N).GT.1.0E-6) GO TO 30
      KF2=1
      F2(N)=1.0E-6
30    CONTINUE
      IF (KF2.EQ.1) CALL DERIV3 (F2,XN,IE,1,F2N)
      IF (KF2.EQ.1) CALL DERIV3 (F2N,XN,IE,1,F2NN)
      IF (NC.NE.0) GO TO 40
C
C       DERIVATIVE CONVERGENCE TEST AT WALL
C
      FPW=(F2(2)-F2(1))/XN(2)
      DIFF=(FPW-FPWW)/FPW
```

AD-A074 972 VIRGINIA POLYTECHNIC INST AND STATE UNIV BLACKSBURG --ETC F/G 20/4
USER'S GUIDE FOR A SYSTEM OF COMPUTER PROGRAMS TO PREDICT FLOWS--ETC(U)
NOV 78 M C HYMAN, S C HOULIHAN, J A HILL
UNCLASSIFIED VPI-AERO-096 NL

3 OF 4
ADA
074972



ADA
074972


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART

```
      FPWW=FPW                                                          MOMO 650
      IF (DIFF.GT.DIF) DIF=DIFF                                         MOMO 660
40    CONTINUE                                                          MOMO 670
      IF (K.GT.1) GO TO 60                                              MOMO 680
      DO 50 N=1,IE                                                      MOMO 690
      F1(N)=F2(N)                                                       MOMO 700
      F1N(N)=F2N(N)                                                     MOMO 710
50    F1NN(N)=F2NN(N)                                                   MOMO 720
60    CONTINUE                                                          MOMO 730
      IF (NC.EQ.0) GO TO 80                                             MOMO 740
      DO 70 N=2,IE                                                      MOMO 750
C                                                                       MOMO 760
C        ALL POINTS CONVERGENCE TEST                                    MOMO 770
C                                                                       MOMO 780
      DIFF=ABS(1.0-FC(N)/F2(N))                                         MOMO 790
      DIFM(N)=DIFF                                                      MOMO 800
      IF (DIFF.GT.DIF) DIF=DIFF                                         MOMO 810
70    CONTINUE                                                          MOMO 820
80    CONTINUE                                                          MOMO 830
      DO 90 N=1,IE                                                      MOMO 840
      FCP(N)=(F2(N)-F1(N))/DS                                           MOMO 850
      FCN(N)=F2N(N)*CRNI+F1N(N)*CCRNI                                   MOMO 860
90    FC(N)=F2(N)*CRNI+F1(N)*CCRNI                                      MOMO 870
      RETURN                                                            MOMO 880
      END                                                               MOMO 890
```

```
      SUBROUTINE PRFILE
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE PRFILE IS CALLED BY MAIN.
C
C     SUBROUTINE PRFILE GENERATES THE INITIAL VELOCITY PROFILE.
C
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),F
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1
     301),XN(102),XN2(102),Y(101),YOVDEL(101),YOVTHT(101),YY(101)
      COMMON /COMWLL/ A1B,E1,FF,F2N1,VW
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLD
     1,XJ,XJFAC,Z,ZOL,RO
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
      COMMON /NMLCRD/ ADTEST,ETAINF,XKETA
      COMMON /REF/ PREF,RHOREF,UREF
      COMMON /CFPR/ CF
      COMMON /VSCSTY/ EPSVD,XK1,XK2
      COMMON /TVCURV/ TVC,BODANG,FTVC(101),RCRW(101),RORWSQ(101),RW
      COMMON /SPTVC/ RNOSE
      COMMON /WAK/ XKFAC
      DATA YES/3HYES/
C
C
      EXP(X)=DEXP(X)
      FLOAT(M)=DFLOAT(M)
C
C      DEVELOP INITIAL ETA GRID
C
```

```
      IF (XKETA.EQ.1.0) DETA1=ETAINF/FLOAT(IE-1)
      IF (XKETA.NE.1.0) DETA1=ETAINF*(XKETA-1.0)/(XKETA**IM-1.0)
      DN(1)=DETA1
      XN(1)=0.00
      Y(1)=0.0
      XN(IE)=ETAINF
      DO 10 N=1,IE
      DN(N+1)=XKETA*DN(N)
      XN(N+1)=XN(N)+DN(N)
10    CONTINUE
      YY(1)=1.D0
      RW=1.D0
      IF (XJFAC.GT.1.D-05.OR.XKFAC.GT.1.D-05) RW=RO
      RTVC(1)=RW
      RORW(1)=1.D0
      IF (XKFAC.GT.1.D-05) RORW(1)=0.D0
      RORWSQ(1)=RORW(1)**2
      IF (NOSE.EQ.1) GO TO 20
C
C      SHARP NOSE
C
      RD=RW
      RDD=RW
      RDDD=RD
      IF (XKFAC.GT.1.D-04) RDDD=UE
      RT=1.D0
      GO TO 30
20    CONTINUE
C
C      BLUNT NOSE
C
      RD=RNOSE
```

```
      RDD=1.DO
      RDDD=RD
      IF (XKFAC.GT.1.D-04) RDDD=DUEDS
      RT=0.DO
30    CONTINUE
C
C     DEVELOP STRETCHED AND UNSTRETCHED COORDINATES FOR NO CURVATURE
C     CASES
C
      CSORW=1.DO
      IF (RW.GT.1.D-06) CSORW=DCOS(BODANG)/RW
      XDOUG1=ETAINF/6.DO
      DO 50 N=1,IF
      Y(N)=PNC*XN(N)
      YY(N)=Y(N)*EPSVD
      RORW(N)=1.DO
      RORWSQ(N)=1.DO
      RTVC(N)=RW
C
C        DEVELOP STRETCHED AND UNSTRETCHED COORDINATES FOR TVC CASES
C
      IF (TVC.NE.YES) GO TO 40
      YY(N)=(-1.DO+DSQRT(1.DO+2.DO*EPSVD*Y(N)*CSORW))/CSORW
      RDUM=RD+RDD*YY(N)*CSORW
      RTVC(N)=RT*RDUM
      RORW(N)=RDUM/RDDD
      RORWSQ(N)=RORW(N)**2
      Y(N)=YY(N)/EPSVD
40    CONTINUE
C
C     USE ASSUMED VELOCITY PROFILE FOR INITIAL GUESS. VARIATION OF 1/6
C     POWER LAW.
```

```
C                                                                       PRF0 970
      F1(N)=1.0-EXP(-XN(N)/XDOUG1)                                      PRF0 980
      F1N(N)=EXP(-XN(N)/XDOUG1)                                         PRF0 990
      IF (F1N(N).LE.1.D-25) F1N(N)=0.0                                  PRF01000
      F1NN(N)=-F1N(N)                                                   PRF01010
      F2NN(N)=F1NN(N)                                                   PRF01020
      FC(N)=F1(N)                                                       PRF01030
      FCN(N)=F1N(N)                                                     PRF01040
      FCP(N)=0.0                                                        PRF01050
      VC(N)=VW-XN(N)/XDOUG1                                             PRF01060
      EPSPL(N)=0.0                                                      PRF01070
      F2(N)=F1(N)                                                       PRF01080
      YOVDEL(N)=0.0                                                     PRF01090
      YOVTHT(N)=0.0                                                     PRF01100
      CHI(N)=0.0                                                        PRF01110
   50 CONTINUE                                                          PRF01120
      F1(IE)=1.0                                                        PRF01130
      F2(IE)=1.0                                                        PRF01140
      FC(IE)=1.0                                                        PRF01150
      F1N(IE)=0.0                                                       PRF01160
      F1NN(IE)=0.0                                                      PRF01170
      RETURN                                                            PRF01180
      END                                                               PRF01190
```

```
      SUBROUTINE READIN
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE READIN CALLS SUBROUTINES WRITE1 AND WRITE2.
C
C     SUBROUTINE READIN IS CALLED BY MAIN AND SUBROUTINE WRITE2.
C
C     SUBROUTINE READIN PROVIDES THE INPUT OF DATA FOR THE PROGRAM.
C
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),F
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1
     301),XN(102),XN2(102),Y(101),YOVDEL(101),YOVTHT(101),YY(101)
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(
     1101),RZ(501)
      COMMON /CNVERG/ CONVRG,CCRNI,CRNI,DIF,DIFF,NC
      COMMON /COEFF/ CFBAR,CFBREX,CFE,CFINF,CFRES,CFREY,DEL,DELST,DELTA,
     1DSTARK,DSTARX,HAFCF,RETHET,REX,THET
      COMMON /COEFF2/ DELORF,DELOX,DSAXOR,DSTODL,DSTORF,DSTOTH,DSTOX,ROR
     1EFL,THODEL,THOREF,XOREFL,ZOREFL
      COMMON /COEFF3/ CF1,SUM,SUM1,SUMT
      COMMON /COMWLL/ A18,E1,FF,F2N1,VW
      COMMON /CFPR/ CF
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO
      COMMON /FRSTRM/ PFS,REINF,RHOFS,UFS
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLD
     1,XJ,XJFAC,Z,ZOL,RO
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
      COMMON /NTEGER/ II,IPFL,IPRINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,L
     1AMTRB,NITTOT
      COMMON /NMLCRD/ ADTEST,ETAINF,XKETA
```

```
      COMMON /REF/ PREF,RHOREF,UREF                                     REA0 330
      COMMON /STAG/ P10,PSTAG,RHO                                       REA0 340
      COMMON /TRANS/ ATR,CHICRT,CHIMAX,GAMMA,XBAR,XIBAR,KTRANS          REA0 350
      COMMON /VSCSTY/ EPSVD,XK1,XK2                                     REA0 360
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW     REA0 370
      COMMON /SPTVC/ RNOSE                                              REA0 380
      COMMON /WTE/ IW,IWT                                               REA0 390
      COMMON /TE/ KTE,KTSTTE                                            REA0 400
      COMMON /WAK/ XKFAC                                                REA0 410
      COMMON /HESS/ IWRTH,ICARD2,ICARD3,ICARD4,NN,IBON,HEDR(10),CASE,KHEREA0 420
     1SS                                                                REA0 430
      COMMON /DRAG/ CDV1,CDV2,XCP,XCP1,CP(501),CP1(501),ROMAX           REA0 440
      COMMON /INTACT/ BETAS                                             REA0 450
      COMMON /STRTUP/ IREAD,IRSRT,KK,ITER                               REA0 460
      COMMON /BDCOOR/ ZDELST(501),RODELS(501),ZOLD(501),ROLD(501),MM    REA0 470
      COMMON /APLDAT/ XLOCA,DYDX,JSTA,NOPRFS                            REA0 480
C                                                                       REA0 490
C                                                                       REA0 500
      DIMENSION LABEL(18)                                               REA0 510
C                                                                       REA0 520
      DIMENSION SUMDS(501)                                              REA0 530
      DATA LST/4HLAST/                                                  REA0 540
C                                                                       REA0 550
C                                                                       REA0 560
      WRITE (6,130)                                                     REA0 570
C                                                                       REA0 580
C                                                                       REA0 590
C     READ IN THE PARAMETERS FOR THE RESTART/ITERATION PROCEEDURE       REA0 600
      READ (5,160) IREAD,IRSRT,ITER,KHESS                               REA0 610
C                                                                       REA0 620
C     IF THERE IS NO RESTART TAPE TO BE READ, INPUT FOR NON ITERATION   REA0 630
C     OPERATION                                                         REA0 640
```

```
C                                                                       REAO 650
      IF (IREAD.EQ.0) GO TO 10                                          REAO 660
      CALL RESTAR                                                       REAO 670
      GO TO 50                                                          REAO 680
10    IF (KHESS.EQ.0) GO TO 20                                          REAO 690
C                                                                       REAO 700
C     FOR RESTART CASE, READ IN STANDARD HESS INPUT                     REAO 710
C                                                                       REAO 720
      READ (5,260) HEDR,CASE,ICARD2,ICARD3,ICARD4,NN,IBDN               REAO 730
      READ (5,210) BETAS,ROMAX                                          REAO 740
20    READ (5,190) LABEL                                                REAO 750
C                                                                       REAO 760
C        TITLE CARD                                                     REAO 770
C                                                                       REAO 780
      WRITE (6,200) LABEL                                               REAO 790
C                                                                       REAO 800
C                                                                       REAO 810
      READ (5,180) NOSE,LAMTRR,KVSLAW,KTRANS,KTRNSN,KONSET,KTE          REAO 820
      READ (5,160) NIT1,NIT2,NIT3,NC                                    REAO 830
      READ (5,140) IE,KEND,IIMAX                                        REAO 840
      READ (5,170) KADETA,KL,IPFL,IPRINT,IWT                            REAO 850
      READ (5,210) CHICRT,XBAR,ATR                                      REAO 860
      READ (5,210) UFS,REINF,PFS,PSTAG,SCF,RNOSE                        REAO 870
      READ (5,210) XK1,XK2,CONVRG,ADTEST                                REAO 880
      READ (5,210) DX,CRN1,XKETA,ETAINF,REFLEN                          REAO 890
      READ (5,150) XJFAC,XKFAC,TVC                                      REAO 900
      READ (5,220) DXMAX,XLOCA,DYDX,JSTA,NCPRFS                         REAO 910
      READ (5,230) (IPR(J),J=1,IPRINT)                                  REAO 920
      READ (5,230) (IPRFL(J),J=1,IPFL)                                  REAO 930
      IWRTH=1                                                           REAO 940
      J=0                                                               REAO 950
30    J=J+1                                                             REAO 960
```

```
      READ (5,240) ZA(J),RZ(J),XSTA(J),PZ(J),LSTC
      IF (LSTC.EQ.LST) GO TO 40
      GO TO 30
40    IIMAX=J
50    J=IIMAX
      IF (KHESS.EQ.0) GO TO 100
      READ (18,250) (CP(I),I=1,J)
C
C     IF ON INITIAL ITERATION, SAVE INITIAL PRESSURE COEFF'S. FOR USE IN
C     DRAG CALCULATIONS.
C
      DO 60 I=1,J
60    CP1(I)=CP(I)
C
C     DEVELOP PRESSURES FROM PRESSURE COEFF'S.
C
      RHO=(PSTAG-PFS)*2/UFS**2
      DO 70 I=1,J
70    PZ(I)=(PFS+CP(I)*0.5*RHO*UFS**2)/PSTAG
      IF (IREAD.NE.1) GO TO 90
      DO 80 I=1,J
      PZ(I)=PZ(I)*P10
80    UEZ(I)=DSQRT(2.D0*(P10-PZ(I)))
      WRITE (6,200) LABEL
      WRITE (6,110) P10
      WRITE (6,120) UE,PE
      CALL WRITE1
      RETURN
90    CONTINUE
      IF (IREAD.EQ.1) RETURN
100   CONTINUE
      IW=IWT-1
```

```
      IF (NIT1.EQ.0) NIT1=3                                             REA01290
      IF (NIT1.LT.0) NIT1=0                                             REA01300
      IF (NIT2.EQ.0) NIT2=6                                             REA01310
      IF (NIT3.EQ.0) NIT3=9                                             REA01320
      IF (KL.EQ.0) KL=2                                                 REA01330
      IF (ADTEST.EQ.0.0) ADTEST=0.0010                                  REA01340
      IF (CONVRG.EQ.0.0) CONVRG=0.0010                                  REA01350
      IF (ETAINF.EQ.0.0) ETAINF=100.0                                   REA01360
      IF (XKETA.EQ.0.0) XKETA=1.09                                      REA01370
      IF (LAMTRB.EQ.0) LAMTRB=1                                         REA01380
      IF (NOSE.EQ.0) NOSE=1                                             REA01390
      IF (XK1.EQ.0.0) XK1=0.40                                          REA01400
      IF (XK2.EQ.0.0) XK2=0.01680                                       REA01410
      IF (KTRNSN.GT.IIMAX) KTRNSN=IIMAX                                 REA01420
      IF (KTRNSN.EQ.0) KTRNSN=IIMAX                                     REA01430
      IF (KTE.EQ.0) KTE=IIMAX                                           REA01440
      IF (KONSET.EQ.0) KONSET=IIMAX                                     REA01450
      RETURN                                                            REA01460
C                                                                       REA01470
C                                                                       REA01480
C                                                                       REA01490
110   FORMAT (1H,,2X,4HP10=,D15.7)                                      REA01500
120   FORMAT (9H     UESO=,D15.7,9H     PESO=,D15.7)                    REA01510
130   FORMAT (60X,13HPROGRAM  ICBL//60X,13HA PROGRAM FOR/21X,89H2-D AND REA01520
     1AXISYMMETRIC INCOMPRESSIBLE, LAMINAR, TRANSITIONAL AND/OR TURBULENREA01530
     2T BOUNDARY-LAYER/42X,47HFLOW INCLUDING EFFECTS OF TRANSVERSE CURVATREA01540
     3URE./66X,2HBY/45X,41HD.L. DWOYER, C.H. LEWIS AND E.C. ANDERSON/47XREA01550
     4,37HAEROSPACE AND OCEAN ENGINEERING DEPT./41X,51HVIRGINIA POLYTECHREA01560
     5NIC INSTITUTE AND STATE UNIVERSITY/56X,21HBLACKSBURG, VA. 24061/56REA01570
     6X,22HPHONE - (703)-951-6126//)                                    REA01580
140   FORMAT (4(7X,I3))                                                 REA01590
150   FORMAT (2F12.6,4X,A3)                                             REA01600
```

```
160   FORMAT (6(8X,I2))                                    REA01610
170   FORMAT (6(7X,I3))                                    REA01620
180   FORMAT (3(8X,I2),4(7X,I3),9X,I1)                     REA01630
190   FORMAT (18A4)                                        REA01640
200   FORMAT (1H0,30X,18A4////)                            REA01650
210   FORMAT (6F12.6)                                      REA01660
220   FORMAT (3F12.6,2I3)                                  REA01670
230   FORMAT (14I5)                                        REA01680
240   FORMAT (4E12.6,A4)                                   REA01690
250   FORMAT (12X,F12.6)                                   REA01700
260   FORMAT (10A6,2X,A6/I3/I5/1X,I4,I5/9X,I1)             REA01710
      END                                                  REA01720
```

```
      SUBROUTINE RESTAR                                                 RESO  10
C                                                                       RESO  20
C                                                                       RESO  30
C     SUBROUTINE RESTAR WRITES A DATA TAPE AT THE RESTART POINT         RESO  40
C     INCLUDING ALL VAGRIABLES THAT ARE NECESSARY FOR THE               RESO  50
C     PROGRAM TO CALCULATE A NEW SOLUTION USING A PREVICUS ITER-        RESO  60
C     ATION SOLUTION.                                                   RESO  70
C                                                                       RESO  80
C                                                                       RESO  90
      IMPLICIT REAL*8(A-H,O-Z)                                          RESO 100
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)RESO 110
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),FRESO 120
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1RESO 130
     301),XN(102),XN2(102),Y(101),YOVDEL(101),YOVTHT(101),YY(101)       RESO 140
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(RESO 150
     1101),RZ(501)                                                      RESO 160
      COMMON /COEFF/ CFBAR,CFBREX,CFE,CFINF,CFRES,CFREY,DEL,DELST,DELTA,RESO 170
     1DSTARK,DSTARX,HAFCF,RETHET,REX,THET                               RESO 180
      COMMON /COEFF2/ DELORF,DELOX,DSAXOR,DSTODL,DSTORF,DSTOTH,DSTOX,RORRESO 190
     1EFL,THODEL,THOREF,XOREFL,ZOREFL                                   RESO 200
      COMMON /COEFF3/ CF1,SUM,SUM1,SUMT                                 RESO 210
      COMMON /COMWLL/ A1B,E1,FF,F2N1,VW                                 RESO 220
      COMMON /CFPR/ CF                                                  RESO 230
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO               RESO 240
      COMMON /FRSTRM/ PFS,REINF,RHOFS,UFS                               RESO 250
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLDRESO 260
     1,XJ,XJFAC,Z,ZOL,RO                                                RESO 270
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NITRESO 280
     13,NOSE                                                            RESO 290
      COMMON /NTEGER/ II,IPFL,IPRINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,LRESO 300
     1AMTRB,NITTOT                                                      RESO 310
      COMMON /TRANS/ ATR,CHICRT,CHIMAX,GAMMA,XBAR,XIBAR,KTRANS,KONSET   RESO 320
```

```
      COMMON /VSCSTY/ EPSVD,XK1,XK2                                     RESO 330
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW     RESO 340
      COMMON /LWALL/ UOUPLS(101),YPLUS(101),UDEF(101),UPLUS             RESO 350
      COMMON /STAG/ P10,PSTAG,RHO                                       RESO 360
      COMMON /CNVERG/ CONVRG,CCRNI,CRNI,DIF,DIFF,NC                     RESO 370
      COMMON /NMLCRD/ ADTEST,ETAINF,XKETA                               RESO 380
      COMMON /REF/ PREF,RHOREF,UREF                                     RESO 390
      COMMON /WTE/ IW,IWT                                               RESO 400
      COMMON /TE/ KTE,KTSTTE                                            RESO 410
      COMMON /WAK/ XKFAC                                                RESO 420
      COMMON /HESS/ IWRTH,ICARD2,ICARD3,ICARD4,NN,IBDN,HEDR(10),CASE,KHERESO 430
     1SS                                                                RESO 440
      COMMON /INTACT/ BETAS                                             RESO 450
      COMMON /DRAG/ CDV1,CDV2,XCP,XCP1,CP(501),CP1(501),ROMAX           RESO 460
      COMMON /BDCOOR/ ZDELST(501),RODELS(501),ZOLD(501),ROLD(501),MM    RESO 470
      COMMON /STRTUP/ IREAD,IRSRT,KK                                    RESO 480
      COMMON /APLDAT/ XLOCA,DYOX,JSTA,NCPRFS                            RESO 490
      IF (IREAD.EQ.1) GO TO 10                                          RESO 500
      IF (IRSRT.GT.1) GO TO 30                                          RESO 510
      GO TO 40                                                          RESO 520
   10 READ (20) AOB,AOBP,DN,EPSPL,F2,F2N,F2NN,FC,FCN,FCP,VC,XN,Y,YOVDEL,RESO 530
     1YOVTHT,YY,CONVRG,CCRNI,CRNI,DIF,DIFF,NC,CFBAR,CFBREX,CFE,CFINF,CFRRESO 540
     2ES,CFREY,DEL,DELST,DELTA,DSTARK,DSTRAX,HAFCF,RETHET,REX,THET,DELCRRESO 550
     3F,DELOX,DSAXOR,DSTODL,DSTORF,DSTOTH,DSTOX,RCREFL,THODEL,THOREF,XORRESO 560
     4EFL,ZORELF,CF1,SUM,A1B,E1,FF,F2N1,VW,CF,DUEDS,PE,PESO,PP,UE,UERO2,RESO 570
     5UESO,PFS,REINF,RHOFS,UFS,DS,DX,DX1,DXMAX,DXOLD                    RESO 580
      READ (20) PNC,RO,REFLEN,SCF,X,XI,XI2,XIOLD,XOLD,XJ,XJFAC,Z,ZOL,IE,RESO 590
     1IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT3,NOSE,II,IPFL,IPRRESO 600
     2INT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,LAMTRB,NITTOT,ADTEST,ETAINF,RESO 610
     3XKETA,P10,PSTAG,ATR,CHICRT,CHIMAX,GAMMA,XBAR,XIBAR,KTRANS,EPSVD,XKRESO 620
     41,XK2,HEDR,CASE,ICARD2,ICARD3,ICARD4,NN,IBDN,BETAS,XSTA,UEZ,PZ,ZA,RESO 630
     5IPR,IPRFL,RZ,KHESS,PREF,RHOREF,UREF,IW,IWT,KTE,KTSTTE,XKFAC,KONSETRESO 640
```

```
     6,BETA,A1,A2,A3,A4,CHI,DN2,EPSO,F1,F1N,F1NN,XN2,TVC,BODANG,RTVC,RORR
     7W,RORWSQ,RW,UOUPLS,YPLUS,UDEF,UPLUS,KK,ZDELST,RODELS,CDV1,CDV2,XCP
     8,XCP1,CP,CP1,SUM1,SUMT,ROMAX,RHC,MM,XLCCA,DYDX,JSTA,NOPRFS
C
C     AFTER READING ALL VARIABLES FROM DATA TAPE, REWIND TO REPLACE
C     DATA FROM CURRENT RUN FOR NEXT ITERATION.
C
      REWIND 20
      READ (3,100) HEDR,CASE,ICARD2,ICARD3,ICARD4,NN,IBDN
C
C     READ IN ALL DISPLACED BODY COORDINATES
C
      READ (3,90) (ZOLD(I),ROLD(I),I=1,NN)
C
C     REWIND AND PLACE AT THE BEGINNING OF THE TAPE THE INPUT PARAMTERS
C     AND THE BODY POINTS (ZDELST,RODELS) UP TO THE RESTART POSITION.
C
      REWIND 3
      WRITE (3,50) HEDR,CASE,ICARD2,ICARD3,ICARD4,NN,IBDN
      WRITE (3,60) (ZDELST(I),RODELS(I),I=1,MM)
C
C     ROLLBACK X VALUE FOR NEXT ITERATION
C
      XOLD=X
      XIOLD=XI
      DXIOLD=DX
      CALL CHANGE
      DO 20 N=1,IE
      F1(N)=F2(N)
      F1N(N)=F2N(N)
      F1NN(N)=F2NN(N)
20    CONTINUE
```

```
      NIT=0                                                             RESO 970
      ISTOP=0                                                           RESO 980
      CALL DELTAS                                                       RESO 990
      UERO2=UE                                                          RESO1000
      IF (XJFAC.GT.1.D-04) UERO2=UE*RO**2                               RESO1010
      IF (XKFAC.GT.1.D-04) UERO2=UE**3                                  RESO1020
      CF1=2.D0*UE*A1B/PNC*EPSVD*DCOS(BODANG)                            RESO1030
      KKMX1=K+10                                                        RESO1040
      KK=K+1                                                            RESO1050
      RETURN                                                            RESO1060
C                                                                       RESO1070
C     WRITE A TAPE WITH NEW SOLUTION DATA                               RESO1080
C                                                                       RESO1090
30    WRITE (20) AOB,AOBP,DN,EPSPL,F2,F2N,F2NN,FC,FCN,FCP,VC,XN,Y,YOVDELRESO1100
     1,YOVTHT,YY,CONVRG,CCRNI,CRNI,DIF,DIFF,NC,CFBAR,CFBREX,CFE,CFINF,CFRESO1110
     2RES,CFREY,DEL,DELST,DELTA,DSTARK,DSTRAX,HAFCF,RETHET,REX,THET,DELORESO1120
     3RF,DELOX,DSAXOR,DSTCOL,DSTORF,DSTOTH,DSTOX,ROREFL,THODEL,THOREF,XORESO1130
     4REFL,ZORELF,CF1,SUM,A1B,E1,FF,F2N1,VW,CF,DUEDS,PE,PESO,PP,UE,UERO2RESO1140
     5,UESO,PFS,REINF,RHOFS,UFS,DS,DX,DX1,DXMAX,DXOLD                   RESO1150
      WRITE (20) PNC,RO,REFLEN,SCF,X,XI,XI2,XICLD,XOLD,XJ,XJFAC,Z,ZOL,IERESO1160
     1,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT3,NOSE,II,IPFL,IPRESO1170
     2RINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,LAMTRB,NITTOT,ADTEST,ETAINFRESO1180
     3,XKETA,P10,PSTAG,ATR,CHICRT,CHIMAX,GAMMA,XBAR,XIBAR,KTRANS,EPSVD,XRESO1190
     4K1,XK2,HEDR,CASE,ICARD2,ICARD3,ICARD4,NN,IBDN,BETAS,XSTA,UEZ,PZ,ZARESO1200
     5,IPR,IPPFL,RZ,KHESS,PREF,PHOREF,UREF,IW,IWT,KTE,KTSTTE,XKFAC,KONSERESO1210
     6T,BETA,A1,A2,A3,A4,CHI,DN2,EPSC,F1,FIN,FINN,XN2,TVC,BODANG,RTVC,RORESO1220
     7RW,RORWSQ,RW,UOUPLS,YPLUS,UDEF,UPLUS,KK,ZDELST,RODELS,CDV1,CDV2,XCRESO1230
     8P,XCP1,CP,CP1,SUM1,SUMT,ROMAX,RHC,MM,XLOCA,DYDX,JSTA,NOPRFS       RESO1240
      WRITE (6,70)                                                      RESO1250
      WRITE (6,80) X,XI,Z,RO,BETA,PP,K                                  RESO1260
      WRITE (21,70)                                                     RESO1270
      WRITE (21,80) X,XI,Z,RO,BETA,PP,K                                 RESO1280
```

```
40    RETURN                                                              RES01290
C                                                                         RES01300
50    FORMAT (10A6,2X,A6,/I3/I5/1X,I4,I5/9X,I1)                           RES01310
60    FORMAT (2F12.6)                                                     RES01320
70    FORMAT (10X,45HRESTART TAPE WRITTEN AT X,X1,Z,RO,BETA,PP,K =)       RES01330
80    FORMAT (10X,6F10.5,7X,I3)                                           RES01340
90    FORMAT (2F12.6)                                                     RES01350
100   FORMAT (10A6,2X,A6/I3/I5/1X,I4,I5/9X,I1)                            RES01360
      END                                                                 RES01370
```

```
      SUBROUTINE SOLVE (W1NN,W1N,W1,W2NN,W2N,W2,E1,F11,CRNI)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE SOLVE CALLS SUBROUTINE DERIV3.
C
C     SUBROUTINE SOLVE IS CALLED BY SUBROUTINE MOMENT.
C
C     SUBROUTINE SOLVE CALCULATES THE SOLUTION FOR A GENERAL PARABOLIC
C     PARTIAL DIFFERENTIAL EQUATION WHEN THE P.D.E. IS REPRESENTED BY A
C     SYSTEM OF IMPLICIT, THREE POINT FINITE DIFFERENCE EQUATIONS.
C
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),F
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1
     301),XN(102),XN2(102),Y(101),YOVDEL(101),YOVTHT(101),YY(101)
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLD
     1,XJ,XJFAC,Z,ZOL,RO
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
C
      DIMENSION W2NN(101), W2N(101), W2(101), W1NN(101), W1N(101), W1(10
     11)
C
      DIMENSION E(102), F(102)
C
      EQUIVALENCE (E(1),DN2(1)), (F(1),XN2(1))
C
C
      E(1)=E1
      F(1)=F11
      DO 10 N=2,IM
      A=(2.00-A1(N)*DN(N))/(DN(N-1)*(DN(N)+DN(N-1)))*CRNI
```

```
      B=((-2.0+A1(N)*(DN(N)-DN(N-1)))/(DN(N)*DN(N-1))+A2(N))*CRNI+A4(N)/
     1DS
      CC=(2.00+A1(N)*DN(N-1))/(DN(N)*(DN(N)+DN(N-1)))*CRNI
      D=-(W1NN(N)+A1(N)*W1N(N)+A2(N)*W1(N))*(1.00-CRNI)-A3(N)+A4(N)*W1(N
     1)/DS
      E(N)=-CC/(B+A*E(N-1))
   10 F(N)=(D-A*F(N-1))/(B+A*E(N-1))
      W2(IE)=1.00
      KON=IM
      DO 20 N=2,IE
      W2(KON)=E(KON)*W2(KON+1)+F(KON)
   20 KON=KON-1
      CALL DERIV3 (W2,XN,IE,1,W2N)
      CALL DERIV3 (W2N,XN,IE,1,W2NN)
      RETURN
      END
```

```
      FUNCTION TLU (NTABLE,Z,X,XSTAR,NFLAG)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     FUNCTION TLU IS CALLED BY SUBROUTINE INTERP.
C
C        FUNCTION TLU IS A ONE-DIMENSIONAL TABLE LOOK-UP PROGRAM.
C        CORRESPONDING VALUES OF X (ALWAYS INCREASING) AND Z ARE STORED
C        IN THE ARRAYS X(1)...X(NTABLE) AND Z(1)...Z(NTABLE).  USING
C        LINEAR INTERPOLATION, THIS FUNCTION WILL GENERATE A VALUE OF Z
C        CORRESPONDING TO A SPECIFIED VALUE OF X = XSTAR.
C
      DIMENSION X(NTABLE), Z(NTABLE)
C
C
C     ..... CHECK TO SEE IF XSTAR LIES WITHIN THE SCOPE OF THE
C              TABULATED VALUES X(1)...X(NTABLE) .....
C
      NFLAG=0
      IF (XSTAR.LT.X(1)) GO TO 10
      IF (XSTAR.LE.X(NTABLE)) GO TO 20
10    NFLAG=1
      TLU=0.0
      RETURN
C
C     ..... SEARCH TO FIND TWO SUCCESSIVE ENTRIES,
C        X(I-1) AND X(I), BETWEEN WHICH XSTAR LIES .....
C
20    I=1
30    IF (X(I).GT.XSTAR) GO TO 40
      IF (I.GE.NTABLE) GO TO 40
      I=I+1
      GO TO 30
```

```
C                                                                       TLUO 330
C      ..... LINEARLY INTERPOLATE TU FIND CORRESPONDING VALUE OF Z ..... TLUO 340
C                                                                       TLUO 350
40     TLU=Z(I-1)+(XSTAR-X(I-1))*(Z(I)-Z(I-1))/(X(I)-X(I-1))            TLUO 360
       RETURN                                                           TLUO 370
C                                                                       TLUO 380
       END                                                              TLUO 390
```

```
      SUBROUTINE VELDAT
C
C       SUBROUTINE VELDAT WRITES OUT THE VELOCITY PROFILE AT THE UP-
C       STREAM EDGE OF THE NAVIER STOKES REGION TO BE USED AS BOUNDARY
C       CONDITIONS FOR THE APLNS PROGRAM.
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /APLDAT/ XLOCA,DYDX,JSTA,NOPRFS
      COMMON /ARRAY1/ A0B(101),A0BP(101),A1(101),A2(101),A3(101),A4(101)
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),F
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1
     301),XN(102),XN2(102),Y(101),YOVDEL(101),YOVTHT(101),YY(101)
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(
     1101),RZ(501)
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLD
     1,XJ,XJFAC,Z,ZCL,RO
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
      COMMON /COEFF/ CFBAR,CFBREX,CFE,CFINF,CFRES,CFREY,DEL,DELST,DELTA,
     1DSTARK,DSTARX,HAFCF,RETHET,REX,THET
C
C
      IF (X.LT.XSTA(JSTA)) RETURN
      DO 10 I=JSTA,IIMAX
      IF (X.EQ.XSTA(I)) GO TO 20
10    CONTINUE
      RETURN
20    XBACK=1.0-XLOCA
      DXI=XBACK/NOPRFS
      DYI=DYDX*DXI
      WRITE (7,40) DYDX,DYI,IE
```

```
      WRITE (7,50) DEL,UE
      DO 30 I=1,IE
30    WRITE (7,50) YY(I),FC(I)
      RETURN
C
40    FORMAT (2F12.6,I3)
50    FORMAT (2F12.6)
      END
```

```
      SUBROUTINE WRITE1
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINE WRITE1 IS CALLED BY MAIN AND SUBROUTINE READIN.
C
C     THIS SUBROUTINE PROVIDES THE CUTPUT OF THE INPUT DATA AND THE
C     INITIALIZED DATA.
C
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPPFL(
     1101),RZ(501)
      COMMON /CNVERG/ CONVRG,CCRNI,CRNI,DIF,DIFF,NC
      COMMON /COMWLL/ A1B,E1,FF,F2N1,VW
      COMMON /CFPR/ CF
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO
      COMMON /FRSTRM/ PFS,REINF,RHOFS,UFS
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLD
     1,XJ,XJFAC,Z,ZOL,RO
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NIT
     13,NOSE
      COMMON /NTEGER/ II,IPFL,IPRINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,L
     1AMTRB,NITTOT
      COMMON /NMLCRD/ ADTEST,ETAINF,XKETA
      COMMON /REF/ PREF,RHOREF,UREF
      COMMON /STAG/ P10,PSTAG
      COMMON /TRANS/ ATR,CHICRT,CHIMAX,GAMMA,XBAR,XIBAR,KTRANS
      COMMON /VSCSTY/ EPSVD,XK1,XK2
      COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW
      COMMON /TE/ KTE,KTSTTE
      COMMON /WAK/ XKFAC
      COMMON /STRTUP/ IREAD,IRSPT,KK,ITER
      DATA YES/3HYES/
C
```

```
C
      WRITE (6,180) ITER
      IF (TVC.EQ.YES) WRITE (6,170)
      WRITE (6,20) NOSE,LAMTRB,KTRANS,KTRNSN,KTE
      WRITE (6,30) NIT1,NIT2,NIT3,NC,IE,KEND,IIMAX
      WRITE (6,40) KADETA,KL,IPFL,IPRINT,KVSLAW
      WRITE (6,50) UFS,REINF
      WRITE (6,60) PFS,RHCFS,PSTAG
      WRITE (6,70) CHICRT,XBAR,ATR
      WRITE (6,80) XK1,XK2
      WRITE (6,90) CONVRG,ADTEST,CRNI,XKETA,ETAINF
      WRITE (6,100) EPSVD,SCF,REFLEN
      WRITE (6,110) XJFAC,XKFAC,BCDANG,DX1,DXMAX
      WRITE (6,120) (IPR(J),J=1,IPRINT)
      WRITE (6,130) (IPRFL(J),J=1,IPFL)
      WRITE (6,140)
      DO 10 J=1,IIMAX
      WRITE (6,150) J,ZA(J),RZ(J),XSTA(J),PZ(J),UEZ(J)
10    CONTINUE
      WRITE (5,160)
      RETURN
C
C
C
20     FORMAT (1H0,2X,7HNOSE = ,I1,11H, LAMTRB = ,I1,11H, KTRANS = ,I1,11
      1H, KTRNSN = ,I3,7H KTE = ,I3)
30     FORMAT (1H0,2X,7HNIT1 = ,I2,9H, NIT2 = ,I2,9H, NIT3 = ,I2,7H, NC =
      1 ,I2,7H, IE = ,I3,9H, KEND = ,I3,10H, IIMAX = ,I2)
40     FORMAT (1H0,2X,9HKADETA = ,I1,7H, KL = ,I1,9H, IPFL = ,I3,11H, IPR
      1INT = ,I3,11H, KVSLAW = ,I1)
50     FORMAT (1H0,2X,6HUFS = ,F12.6,10H, REINF = ,1PE14.7)
60     FORMAT (1H0,2X,6HPFS = ,F12.4,10H, RHCFS = ,1PE13.6,10H, PSTAG = ,
```

```
      1OPF12.4)                                                         WROO 650
70    FORMAT (1HO,2X,10HCHICRIT = ,F12.3,9H, XBAR = ,F12.6,8H, ATR = ,F1WROO 660
     12.6)                                                              WROO 670
80    FORMAT (1HO,2X,6HXK1 = ,F7.4,8H, XK2 = ,F9.6)                     WROO 680
90    FORMAT (1HO,2X,9HCONVRG = ,F7.4,11H, ADTEST = ,F7.4,9H, CRNI = ,F7WROO 690
     1.4,10H, XKETA = ,F7.4,11H, ETAINF = ,F7.2)                        WROO 700
100   FORMAT (1HO,2X,8HEPSVD = ,F12.6,17H, SCALE FACTOR = ,F9.5,21H, REFWROO 710
     1ERENCE LENGTH = ,F9.4)                                            WROO 720
110   FORMAT (1HO,2X,8HXJFAC = ,F7.4,2X,8HXKFAC = ,F7.4,10H, ANGLE = ,F9WROO 730
     1.6,7H DX = ,,F7.4,10H, DXMAX = ,F7.4)                             WROO 740
120   FORMAT (1HO,3X,3HIPR,4X,20I5,//(11X,20I5/))                       WROO 750
130   FORMAT (1HO,3X,5HIPRFL,2X,20I5,//(11X,20I5/))                     WROO 760
140   FORMAT (1HO,3X,14HAXIAL POSITION,14X,2HZA,13X,2HRZ,12X,4HXSTA,10X,WROO 770
     12HPZ,13X,3HUEZ/)                                                  WROO 780
150   FORMAT (9X,I3,10X,5F15.6)                                         WROO 790
160   FORMAT (1HO)                                                      WROO 800
170   FORMAT (43X,46HTHIS RUN INCLUDES TRANSVERSE CURVATURE EFFECTS////)WROO 810
180   FORMAT (1HO,35HI T E R A T I O N    N U M B E R   ,I2)           WROO 820
      END                                                               WROO 830
```

```
      SUBROUTINE WRITE2                                                 WRIO  10
      IMPLICIT REAL*8(A-H,O-Z)                                          WRIO  20
C                                                                       WRIO  30
C     SUBROUTINE WRITE2 CALLS SUBROUTINE READIN.                        WRIO  40
C                                                                       WRIO  50
C                                                                       WRIO  60
C     SUBROUTINE WRITE2 IS CALLED BY MAIN AND SUBROUTINE READIN.        WRIO  70
C                                                                       WRIO  80
C     THIS SUBROUTINE CONTROLS THE OUTPUT AFTER A CONVERGED SOLUTION IS WRIO  90
C     OBTAINED.                                                         WRIO 100
C     COMPLETE SOLUTIONS ARE WRITTEN ONLY AT SELECTED VALUES OF X,      WRIO 110
C     I. E. AT X=XSTA(IPRFL), AND AT STATIONS INCREMENTED BY IWT.       WRIO 120
C     ALL OTHER COMPUTED RESULTS ARE WRITTEN FOR EVERY VALUE OF X AT    WRIO 130
C     WHICH A SOLUTION IS OBTAINED.                                     WRIO 140
C                                                                       WRIO 150
      COMMON /ARRAY1/ AOB(101),AOBP(101),A1(101),A2(101),A3(101),A4(101)WRIO 160
     1,CHI(101),DN(102),DN2(102),EPSO(101),EPSPL(101),F1(101),F1N(101),FWRIO 170
     21NN(101),F2(101),F2N(101),F2NN(101),FC(101),FCN(101),FCP(101),VC(1WRIO 180
     301),XN(102),XN2(102),Y(101),YOVDEL(101),YOVTHT(101),YY(101)       WRIO 190
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(WRIO 200
     1101),RZ(501)                                                      WRIO 210
      COMMON /COEFF/ CFBAR,CFBREX,CFE,CFINF,CFRES,CFREY,DEL,DELST,DELTA,WRIO 220
     1DSTARK,DSTARX,HAFCF,RETHET,REX,THET                               WRIO 230
      COMMON /COEFF2/ DELORF,DELOX,DSAXOR,DSTODL,DSTORF,DSTOTH,DSTOX,RORWRIO 240
     1EFL,THODEL,THOREF,XOREFL,ZOREFL                                   WRIO 250
      COMMON /COEFF3/ CF1,SUM,SUM1,SUMT                                 WRIO 260
      COMMON /COMWLL/ A1B,E1,FF,F2N1,VW                                 WRIO 270
      COMMON /CFPR/ CF                                                  WRIO 280
      COMMON /EDGPRP/ BETA,DUEDS,PE,PESO,PP,UE,UERO2,UESO               WRIO 290
      COMMON /GEOME/ DS,DX,DX1,DXMAX,DXOLD,PNC,REFLEN,SCF,X,XI,XI2,XIOLDWRIO 300
     1,XJ,XJFAC,Z,ZOL,RO                                                WRIO 310
      COMMON /INTGR/ IE,IIMAX,IM,IPRNT,ISTOP,KL,KVSLAW,NIT,NIT1,NIT2,NITWRIO 320
```

```
      13,NOSE                                                          WRIO 330
       COMMON /NTEGER/ II,IPFL,IPRINT,JJ,K,KADETA,KEND,KEP,KSTOP,KTRNSN,LWRIO 340
      1AMTRB,NITTOT                                                    WRIO 350
       COMMON /TRANS/ ATR,CHICRT,CHIMAX,GAMMA,XBAR,XIBAR,KTRANS         WRIO 360
       COMMON /VSCSTY/ EPSVD,XK1,XK2                                   WRIO 370
       COMMON /STAG/ P10,PSTAG,RHO                                     WRIO 380
       COMMON /REF/ PREF,RHOREF,UREF                                   WRIO 390
       COMMON /FRSTRM/ PFS,REINF,RHOFS,UFS                             WRIO 400
       COMMON /TVCURV/ TVC,BODANG,RTVC(101),RORW(101),RORWSQ(101),RW    WRIO 410
       COMMON /WTE/ IW,IWT                                             WRIO 420
       COMMON /LWALL/ UOUPLS(101),YPLUS(101),UDEF(101),UPLUS            WRIO 430
       COMMON /TE/ KTE,KTSTTE                                          WRIO 440
       COMMON /HESS/ IWRTH,ICARD2,ICARD3,ICARD4,NN,IBDN,HEDR(10),CASE,KHEWRIO 450
      1SS                                                              WRIO 460
       COMMON /BDCCOR/ ZDELST(501),RODELS(501),ZCLD(501),RCLD(501),MM   WRIO 470
       COMMON /STRTUP/ IREAD,IRSRT,KK,ITER                             WRIO 480
       COMMON /DRAG/ CDV1,CDV2,XCP,XCPI                                WRIO 490
       COMMON /WKPRL/ RSTAR                                            WRIO 500
C                                                                      WRIO 510
       DIMENSION TPR(102), XMAL(102), PITOT(101), PITOTR(101)           WRIO 520
       DIMENSION VORT1(101)                                            WRIO 530
C                                                                      WRIO 540
       EQUIVALENCE (TPR(1),DN2(1))                                     WRIO 550
       EQUIVALENCE (XMAL(1),XN2(1)), (PITOT(1),EPSO(1)), (PITOTR(1),A1(1)WRIO 560
      1)                                                               WRIO 570
C                                                                      WRIO 580
C                                                                      WRIO 590
       ABS(X)=DABS(X)                                                  WRIO 600
       IF (KHESS.EQ.0) GO TO 30                                        WRIO 610
C                                                                      WRIO 620
C                                                                      WRIO 630
C        WF IS WEIGHTING FACTOR FOR DAMPING BODY COOR ITERATIONS       WRIO 640
```

```
      WF=0.5D0                                                          WRIO 650
      IF (IWRTH.EQ.0) GO TO 30                                          WRIO 660
      IF (K.GE.2) GO TO 10                                              WRIO 670
      WRITE (3,290) HEDR,CASE,ICARD2,ICARD3,ICARD4,NN,IBDN              WRIO 680
      MM=0                                                              WRIO 690
10    THETA=BODANG                                                      WRIO 700
      MM=MM+1                                                           WRIO 710
      RODELS(MM)=RO+DELST*DCOS(THETA)                                   WRIO 720
      ZDELST(MM)=Z-DELST*DSIN(THETA)                                    WRIO 730
      IF (ITER.EQ.1) GO TO 20                                           WRIO 740
      ZDELST(MM)=(1.D0-WF)*ZOLD(MM)+WF*ZDELST(MM)                       WRIO 750
      RODELS(MM)=(1.D0-WF)*ROLD(MM)+WF*RODELS(MM)                       WRIO 760
20    WRITE (3,280) ZDELST(MM),RODELS(MM)                               WRIO 770
      IWRTH=0                                                           WRIO 780
30    XBLANK=0.0                                                        WRIO 790
      WRITE (6,130) X,XI,Z,RO,BETA,PP,NIT,K                             WRIO 800
      WRITE (6,140) UE,PE,DUEDS,CF,EPSVD,KEP,NITTOT                     WRIO 810
      WRITE (6,150) F2N(1),F2N(3),F2NN(1),F2NN(3),UPLUS                 WRIO 820
      IF (X.EQ.XSTA(1)) GO TO 60                                        WRIO 830
      WRITE (6,160) CHIMAX,GAMMA,XIBAR                                  WRIO 840
      WRITE (6,170) REX                                                 WRIO 850
      IF (KTSTTE.GT.0) GO TO 40                                         WRIO 860
      WRITE (6,180) CFE,CFINF,CFBAR,CFBREX,CFREY                        WRIO 870
      TAUW=CFINF*RHOFS*UFS**2/2.0/144.0                                 WRIO 880
      WRITE (6,190) HAFCF,TAUW                                          WRIO 890
40    CONTINUE                                                          WRIO 900
      WRITE (6,200) XOREFL,ZCREFL,ROREFL,DELCX,DSTOX                    WRIO 910
      WRITE (6,210) THOREF,DSTORF,DELORF,DSAXCR                         WRIO 920
      WRITE (6,220) THODEL,DSTODL,DSTOTH,DSTRAX                         WRIO 930
      WRITE (6,230) DSTARK,DELST,DEL,THET,RETHET                        WRIO 940
      ANG=BODANG                                                        WRIO 950
      BDYANG=ANG*180.0D0/DARCOS(-1.0D0)                                 WRIO 960
```

```
      WRITE (6,270) BDYANG                                              WRIO 970
      IF (KHESS.EQ.0) GO TO 50                                          WRIO 980
      WRITE (6,300) SUM,SUM1,SUMT,XCP,XCP1                              WRIO 990
50    IF (KTSTTE.LT.1) GO TO 60                                         WRI01000
      WRITE (6,310) RSTAR                                               WRI01010
60    CONTINUE                                                          WRI01020
      REFF=R0+DELST                                                     WRI01030
      IF (K.EQ.1) GO TO 80                                              WRI01040
      DO 70 NM=1,IPRINT                                                 WRI01050
      IF (X.EQ.XSTA(IPRFL(NM))) GO TO 80                                WRI01060
70    CONTINUE                                                          WRI01070
      IF (ABS(1.0-X/XSTA(KTRNSN)).LE.1.E-6) GO TO 80                    WRI01080
      IW=IW+1                                                           WRI01090
      IF (IW.LE.IWT) GO TO 120                                          WRI01100
      IW=1                                                              WRI01110
80    CONTINUE                                                          WRI01120
      UFAC=UE*UFS                                                       WRI01130
      DO 90 N=1,IE                                                      WRI01140
      PITOT(N)=P10+.5D0*UE**2*(F2(N)**2-1.D0)                           WRI01150
90    CONTINUE                                                          WRI01160
      WRITE (6,240)                                                     WRI01170
      DO 100 N=1,IE,KL                                                  WRI01180
100   WRITE (6,250) XN(N),YY(N),YOVTHT(N),YOVDEL(N),FC(N),FCN(N),YPLUS(NWRI01190
     1),UOUPLS(N),UDEF(N),EPSPL(N),PITOT(N),N                           WRI01200
      WRITE (6,260)                                                     WRI01210
      IF (X.LT.XSTA(N).OR.X.GT.XSTA(N)) GO TO 120                       WRI01220
      DO 110 I=1,IE                                                     WRI01230
110   VORT1(I)=FIN(I)*RTVC(I)*UF**2/DSQRT(2.D0*XI)                      WRI01240
      WRITE (10) (VORT1(I),YY(I),I=1,IE)                                WRI01250
120   CONTINUE                                                          WRI01260
      IF (K.NE.1) IPRNT=IPRNT+1                                         WRI01270
      IF (X.EQ.XSTA(IRSRT)) CALL RESTAR                                 WRI01280
```

```
      RETURN                                                            WRI01290
C                                                                       WRI01300
C                                                                       WRI01310
C                                                                       WRI01320
130   FORMAT (1H0,2X,3HS =,F10.5,1H,/16X,6H  XI =,E10.4,5H, Z =,F9.5,6H,WRI01330
     1 RO =,F8.5,8H, BETA =,F9.5,6H, PP =,F12.5,8H, NIT = ,I3,5H, K =,I3WRI01340
     2)                                                                 WRI01350
140   FORMAT (1H0,2X,4HUE =,F8.5,6H, PE =,F11.5,9H, DUEDS =,F11.5,6H, CFWRI01360
     1 =,F11.5,9H, EPSVD =,E13.6,7H, KEP =,I3,10H, NITTOT =,I5)         WRI01370
150   FORMAT (1H0,2X,8HF2N(1) =,F12.6,10H, F2N(3) =,F12.6,11H, F2NN(1) =WRI01380
     1,F12.6,11H, F2NN(3) =,F12.6,10H, UPLUS = ,E13.6)                  WRI01390
160   FORMAT (1H0,2X,8HCHIMAX =,1PE13.6,9H, GAMMA =,E13.6,9H, XIBAR =,E1WRI01400
     13.6)                                                              WRI01410
170   FORMAT (1H0,2X,5HREX =,1PE12.5)                                   WRI01420
180   FORMAT (1H0,2X,5HCFE =,1PE13.6,9H, CFINF =,E13.6,13H, CF(TOTAL) =,WRI01430
     1E13.6,19H, CFBAR*SQRT(REX) =,E13.6,19H, CFINF*SQRT(REX) =,E13.6)  WRI01440
190   FORMAT (1H0,2X,9HCFINF/2 =,1PE13.6,8H, TAUW =,E13.6,7H P.S.I.)    WRI01450
200   FORMAT (1H0,2X,10HX/REFLEN =,1PE13.6,12H, Z/REFLEN =,E13.6,13H, ROWRI01460
     1/REFLEN =,E13.6,11H, DELTA/X =,E13.6,12H, DELSTR/X =,E13.6)       WRI01470
210   FORMAT (1H0,2X,14HTHETA/REFLEN =,1PE13.6,17H, DELSTR/REFLEN =,E13.WRI01480
     16,16H, DELTA/REFLEN =,E13.6,20H, DELSTRAXI/REFLEN =,E13.6)        WRI01490
220   FORMAT (1H0,2X,13HTHETA/DELTA =,1PE13.6,16H, DELSTR/DELTA =,E13.6,WRI01500
     116H, DELSTR/THETA =,E13.6,13H, DELSTRAXI =,E13.6)                 WRI01510
230   FORMAT (1H0,2X,9HDELSTRK =,1PE12.5,10H, DELSTR =,E12.5,9H, DELTA =WRI01520
     1,E12.5,9H, THETA =,E12.5,11H, RETHETA =,E12.5)                    WRI01530
240   FORMAT (1H0,/5X,3HETA,8X,1HY,7X,7HY/THETA,4X,7HY/DELTA,5X,6HF=U/UEWRI01540
     1,5X,5HFP(N),6X,5HYPLUS,6X,7HU/UPLUS,5X,4HUDEF,6X,4HEPS+,8X,5HPITOTWRI01550
     2,6X,1HN/)                                                         WRI01560
250   FORMAT (F11.6,10F11.6,I4)                                         WRI01570
260   FORMAT (1X,//)                                                    WRI01580
270   FORMAT (1H0,2X,19HLOCAL BODY ANGLE = ,E12.6///)                   WRI01590
280   FORMAT (2F12.6)                                                   WRI01600
```

```
290    FORMAT (10A6,2X,A6/I3/I5/1X,I4,I5/9X,I1)                       WRI01610
300    FORMAT (3X,8H   CDF =,F12.6,8H   CDP =,F12.6,8H   CDT =,F12.6,7H  WRI01620
     1 CP =,F12.6,8H   CP1 =,F12.6)                                     WRI01630
310    FORMAT (10X,7HRSTAR =,F12.6)                                     WRI01640
       END                                                              WRI01650
```

```
      SUBROUTINE ZRO (JJ,X2,Z,RO)
      IMPLICIT REAL*8(A-H,O-Z)
C
C     SUBROUTINF ZRO CALLS SUBROUTINE INTER5
C
C     SUBROUTINE ZRO IS CALLED BY SUBROUTINES BLUNT2 AND CONE2.
C
C     SUBROUTINE ZRO CALCULATES FOR AXISYMMETRIC FLOW AN AXIAL DISTANCE
C     Z AND A RADIUS RO CORRESPONDING TO A SURFACE DISTANCE X2.
C
      COMMON /ARRAY2/ PZ(501),UEZ(501),XSTA(501),ZA(501),IPR(101),IPRFL(
     1101),RZ(501)
C
C
      J=0
10    J=J+1
      IF (X2.GT.XSTA(J)) GO TC 10
      IF (J.LT.3) J=3
      IF (J.GT.(JJ-2)) J=JJ-2
      CALL INTER5 (X2,XSTA(J-2),XSTA(J-1),XSTA(J),XSTA(J+1),XSTA(J+2),RZ
     1(J-2),RZ(J-1),RZ(J),RZ(J+1),RZ(J+2),RO)
      CALL INTER5 (X2,XSTA(J-2),XSTA(J-1),XSTA(J),XSTA(J+1),XSTA(J+2),ZA
     1(J-2),ZA(J-1),ZA(J),ZA(J+1),ZA(J+2),Z)
      RETURN
      END
```

TABLE 4.1 FIELD EXEC

```
&CONTROL ALL
GLOBAL TXTLIB FORTLIB
FI 06 DISK HESS COORDATA B (LRECL 80 BLKSIZE 80  RECFM FB)
FI 07 TERM
FI 04 DISK INTER DATA B (LRECL 80 BLKSIZE 80 RECFM FB)
FI 05 DISK HESSBLI DATA E  (LRECL 80 BLKSIZE 80 RECFM FB)
FI 02 DISK HESS PARMDATA E (LRECL 80 BLKSIZE 80 RECFM FB
LOAD DATFIELD (NOMAP
START
```

TABLE 4.2 DEVELOP EXEC

```
&CONTROL ALL
&LOOP -LABEL &1
EXEC FIELD
EXEC HESSBLI HESS COORDATA B
```

-LABEL &CONTINUE

TABLE 4.3 DATFIELD FOFTRAN

```
      DIMENSION X(30,60), Y(30,60), HEDR(15), XB(501), YB(501)
C
C       READ IN  DISPLACED BODY COORDINATES
C
      READ (5,10) HEDR, CASE, ICARD2, ICARD3, ICARD4, NN, IBDN
C     READ(5,80)
      DO 6 I=1, NN
    6 READ (5,20) XB(I),YB(I)
      DATA  NB/1/,NNU/0/,NCF/0/,NAXI/1/,NOFF/1/,IGEOMF/0/,ISIGF/1/,ICURV
     1N/0/,NONEWF/0/,IFORMT/1/,IIBDN/0/,NN1/360/
C
C      WRITE OUT HESS DATA SET FOR EXECUTION WITH OFF BODY POINTS
C
      WRITE (6,70) HEDR, CASE
      WRITE (6,30) NB,NNU,NAXI,NCF,NOFF
      WRITE (6,30)
      WRITE (6,40) IGEOMF,ISIGF,ICURVN,NONEWF,IFORMT,NN
      WRITE (6,50) IBDN
      WRITE (6,20) (XB(I),YB(I),I=1,NN)
      WRITE (6,40 ) IGEOMF,ISIGF,ICURVN,NONEWF,IFORMT,NN1
      WRITE (6,50) IIBDN
```

```
C
C       READ IN THE ITERATION DATA
C
      READ  (4,40) IHESSI
      READ  (4,60) XLCCAT,DX,DY
C
C      M IS THE UPPER MCST ROW CF OFF BODY PCINTS
C      ITERATION.   K IS THE BOTTOM MOST ROW.
C
      M = IHESSI*6
      K = M-5
      YS = DY*(K-1)
      DO 1 I=K,M
      XS = XLOCAT
      DO 3 J=1,60
      Y(I,J) = YS
      X(I,J) = XS
      WRITE (6,20) X(I,J), Y(I,J)
    3 XS = XS+DX
    1 YS = YS+DY
      IHESSI = IHESSI+1
C
C      REWIND FILE WITH REPITITI
C
      REWIND 4
      WRITE (4,40) IHESSI
      WRITE (4,60) XLOCAT,DX,DY
   10 FORMAT (15A4,2X,A4/I3/I5/1X,I4,I5/9X,I1)
   20 FORMAT (2F12.6)
   30 FORMAT (5I1)
   40 FORMAT (5I1,I5)
   50 FORMAT (9X,I1)
```

```
60 FORMAT (3F12.6)                                                DAT00540
70 FORMAT (15A4,2X,A4)                                            DAT00550
90 FORMAT(/)                                                      DAT00560
   STOP                                                           DAT00570
   END                                                            DAT00580
```

TABLE 5.1 APLNS EXEC

```
&CONTROL ALL
GLOBAL TXTLIB FORTXLIB
FI 5 DISK &1 &2 &3
FI 08 DISK STREAM DATA E (RECFM FB BLOCK 87 LRECL 87
FI 11 DISK APLNS VELDATA1 E
TESTBAT
&RC = &RETCODE
&IF &RC EQ 0 FI 06 DISK APLNS OUTPUT E
&IF &RC NE 0 FI 06 PRINTER
LOAD NSTIDY PLTCAPLN ( CLEAR NOMAP
START
```

TABLE 5.2 APLNS FORTRAN

```
      CALL DRFAX6
      STOP
      END
```

```
      SUBROUTINE WORK6(TIME, TS, TTS )
C     AXI-SYM & CORNER                         05/12/76      FAVIN           01000
C     WITHIN WORK6 THE CALCULATIONS ARE PERFORMED WHICH COM-
C     UTE XIW AND XI, THE VALUES OF DU/DY AT THE WALL AND
C     AT ANY POINT IN THE FLOW, RESPECTIVELY. THESE VALUES
C     ARE COMPARED IN SUBROUTINE DRFAX6. THE VALUES FOR XIW
C     AND XI ARE CALCULATED FROM STREAM FUNCTION CALCULATED
C     WITHIN THE GRID REGION.
      IMPLICIT REAL*8 (A-H,O-Z)                                              01001
C
C          U1 -->
C     J=L  +-------------------------------------+
C          |B      *                            C|
C          |       *                             |
C  J=JDEL  |       *                             |
C          .       *                             .
C  Y       .       *                             .        W O R K 6
C          .      *.                             .
C     J=L1 +.....*....+                          |
C          |    *     |                          |
C  J=JCI   * F        | FX                       |
C          |          |                          |                       1014
C          |          |                          |                      01015
C     J=1  |A         |E                        D|                      01016
C          +----------+--------------------------+
C         I=1        I=M1                       I=M
C                            X
C
C                                                                           01002
      REAL*8  K1, KM, KL, KU
      COMMON /REALS/  R1,C7,DX,DY,DT,U2,RHO,EPS,FX,FXRHO,X,Y,U0,
     *  DT2,DTA,TDY,TDX,DY2,  DFRODR,  TRAMP,TOL, K1, KM, KL, KU,
```

```
*   EPG, THRUST
 COMMON /ARRAYS/  PSI,XI,V,U,VA,UA,PSIB,XIB,XIW, EPA(60), U1(60),
*  U10(60), PSI0(60)
 COMMON /INTEGS/  L,L1,JDEL,M,M1,NTS,NFR,PSKP,IDIMY,JDIMY,JCI,IO
 COMMON /IRRAYS/  LAR(60)
 COMMON  /LOGICS/  LSWH1, PRNT, ON, OFF, RSTART, SAVE, SWHFX,
*   ZSTART, CSTART
C     DIMENSIONS ARE SET UP FOR MAXIMUMS OF  L=30, M=60, L1=15 & M1=15      01021
C                                                                            01022
 DIMENSION PSI(60,30), PSIB(60,30), XI(60,30), V(60,30), U(60,30)           01023
 DIMENSION XIA(60,30), XIB(60,30),  VA(60,30), UA(60,30)                    01024
 DIMENSION A(1624,3), B(1624), XL(3248)                                     01025
 DIMENSION          XIW(15), AA(30),BB(30),CC(30),  YY(30,60)               01026
 DIMENSION U0(30), Y(30), X(60), FX(15), FXRHC(15),  DFRODR(15)             01027
 INTEGER  TS, TTS, PSKP                                                     01028
 COMMON IHDR                                                                01029
 INTEGER JHDR(6)                                                            01030
 DATA  JHDR     / 'OPSI','O XI', 'O  V', 'O  U', 'OXIA', 'OXIB' /           01031
 DATA IPSIB /4HOPSB/
 LOGICAL  LSWH1, PRNT, ON, OFF, SWHFX
 LOGICAL  RSTART, SAVE, ZSTART, CSTART
 DIMENSION PSWI(15,15), PSW(15), AI(15), A$(15,15)
 EQUIVALENCE (A(1),A$(1))
C                                                                            01035
 DATA ZERO,ONE,TWO,THREE,FOUR/0.0D0,1.0D0,2.0D0,3.0D0,4.0D0/               01037
C                                                                            01038
C DATA IA,IB,IDIMY,JDIMY  / 1624, 1624, 60, 30 /
 DATA IA,IB  / 1624, 1624 /
 DATA  NLC,NUC, MB, IJOB  / 1,1,1,0 /
 DATA  INCE / 0  /
 DLOG2(X) = DLOG(X) / DLG2                                                  01039
C                                                                            01041
```

```
99 FORMAT(2X,1P10G12.4)
   IF(INCE .EQ. 1)  GO TO 1000
   PRINT 1
 1 FORMAT('0**01/20/77     W O R K 6    WITH EPO')
   INCE=1
C
C     EPO = 18.900D-05
   PRINT 6, EPO
 6 FORMAT(' EPO = ',1PG16.7)
C
   DLG2 = DLOG(TWO)                                                 01040
   M1P1 = M1+1                                                      01085
   MM1 = M-1                                                        01086
   MM2 = M-2                                                        01087
   MM3 = M-3                                                        01088
   LM1 = L-1                                                        01089
   LM2 = L-2                                                        01090
   LM3 = L-3                                                        01091
   L1M1 = L1 - 1                                                    01092
   L1M2 = L1 - 2                                                    01093
   N = LM2*MM2                                                      01094
   R = LM2                                                          01095
   IDIMW = 2*LM2 + 4*MM2 + MM2*(DLOG2(R) + 1)  +1                   01096
43 FORMAT('0HERE IS N, IDIMY AND IDIMW (MAX DIM FOR WW)',/3I8)      01097
   PRINT 43, N, IDIMY, IDIMW                                        01098
   DT2 = DT / TWO                                                   01099
   DX2 = DX*DX                                                      01100
   FDX2 = FOUR * DX2
   DY2 = DY*DY                                                      01101
   DY3 = DY2 * DY                                                   01102
   RYX2 = DY2/DX2                                                   01103
   RXY2 = DX2 / DY2                                                 01104
```

```
      TDY = TWO * DY                                                    01105
      TDX = TWO * DX                                                    01106
      DO 45 I=1,60
      DO 45 J=1,30
      XIA(I,J)=ZERO
      XIB(I,J)=ZERO
      PSIB(I,J)=ZERO
   45 CONTINUE
      DO 44 J=1,L
   44 XIA(1,J)=XI(1,J)
C     XIW = 0;       /*  XI AT THE WALL FOR I=2 TO M1                */ 01107
      DO 50 I=1,M1                                                      01108
   50 XIW(I) = ZERO                                                     01109
C                                                                       01110
      DXT = DX2 / TDY                                                   01135
      DO 51 J=1,LM1                                                     01136
      T = DXT / Y(J+1)                                                  01137
      AA(J) = RXY2 + T                                                  01138
      CC(J) = RXY2 - T                                                  01139
   51 BB(J) = -TWO * RXY2                                               01140
      AA(LM1) = AA(LM1) + CC(LM1)
      CC(LM1) = ZERO
C                                                                       01141
      N$ = JCI-2
      IA$ = 15
      IDGT = 0
      M1M1 = M1-1
      DO 40 I=2,M1M1
C     PRINT 2, I
    2 FORMAT('0I= ',I4,': PSWI(K,I),K=2,M1M1')
         DO 41 IP=1,LM1
         DO 41 JP=1,MM2
```

```
   41    YY(IP,JP) = ZERO
      LJ = LAR(I)-1
      IF(LJ .LE. 0) GO TO 40
C
      YY(LJ,I-1) = ONE
C     IHDR = MASK
C     IF(I .EQ. 4)  CALL DMTD1D(YY, LM1, MM2, L, M)
      CALL PCIS2(MM2,LM1,AA,BB,CC,JDIMY,YY,XL)
C     IF(I .EQ. 4)  CALL DMTD1D(YY, LM1, MM2, L, M)
C
         DO 42 K=2,M1M1
         LJ = LAR(K)-1
         IF(LJ .LE. 0) GO TO 40
   42    PSWI(K,I)  =   YY(LJ, K-1)
C     PRINT 99, (PSWI(K,I),K=2,M1M1)
   40 CONTINUE
 1000 CONTINUE
      NC = 0                                                            01277
         DO 88 J= 2,LM1
      DO 87 I=2,MM1
      EDD = EPA(I)/DX2
      EDC = (EPA(I+1)-EPA(I-1))/FDX2
      LJ = LAR(I) + 1
         IF(J .LT. LJ) GO TO 87
         NC = NC+1                                                      01282
         A(NC,3) = U(I,J)/ TDX    - EDD - EDC                           01283
         A(NC,2) = TWO/DT + (TWO*EPA(I))/DX2                            01284
         A(NC,1) = -U(I,J)/ TDX    - EDD + EDC                          01285
         B(NC)   = EPA(I)*((XI(I,J+1)-2*XI(I,J)+XI(I,J-1))/DY2          01286
     *         + (XI(I,J+1)-XI(I,J-1))/TDY/Y(J)-XI(I,J)/Y(J)**2)        01287
     *         - V(I,J) * (XI(I,J+1)-XI(I,J-1))/TDY                     01288
     *         + XI(I,J) / DT2 + V(I,J) * XI(I,J) / Y(J)                01289
```

```
C
      B(NC) = B(NC)  - (EPA(I+1)-EPA(I-1))/TDX * (
     1  (V(I,J+1) - TWO*V(I,J) + V(I,J-1))/DY2  +
     2  (V(I,J+1) - V(I,J-1)) / TDY / Y(J)  +
     3  (V(I+1,J) - TWO*V(I,J) + V(I-1,J))/DX2 - V(I,J)/Y(J)**2)  -
     4  (EPA(I+1) - TWO*EPA(I) + EPA(I-1))/DX2    *    (
     5  (U(I,J+1)-U(I,J-1))/TDY + (V(I+1,J)-V(I-1,J))/TDX  )
C
C     ********  FOR EPS( Y )  *********
      IF(EPO .EQ. ZERO) GO TO 93
      IF( I .GT. M1) GO TO  93
      IF( J .NE. LJ) GO TO  93
C
      B(NC) = B(NC) + (EPA(I)-EPO)/TDY * ( (U(I,J+1)-TWO*U(I,J)+
     1   U(I,J-1))/DY2 + (U(I,J+1)-U(I,J-1))/TWO/DY2/(LJ-1)  +
     2   (U(I+1,J)-TWO*U(I,J)+U(I-1,J))/DX2 +
     3   (XI(I,J+1)-XI(I,J-1))/TDY
     4   ) +
     5   (EPA(I)-TWO*EPA(I)+EPO)/DY2 *
     6   ( (U(I,J+1)-U(I,J-1))/TDY + (V(I+1,J)-V(I-1,J))/TDX )
     7   + (EPO-EPA(I))/DY/DX *
     8   ( (U(I+1,J)-U(I-1,J))/TDX - (V(I,J+1)-V(I,J-1))/TDY )
   93 CONTINUE
C
C  DETERMINATION OF THE LOCATION OF THE PROPELLER(S)
C
C     IF((I.NE.M1).AND.(I.NE.M1P1)) GO TO 94
      IF(I.NE.M1) GO TO 94
         IF(J .LE. L1) B(NC) = B(NC) + DFRCDR(J)                   01291
   94 IF(I .NE. 2) GO TO 95                                        01292
C     ALONG    AB
               B(NC) = B(NC) - A(NC,1)*XI(1,J)                     01293
```

```
             A(NC,1) = ZERO                                             01294
   95 CONTINUE                                                          01295
      IF( I .NE. MM1) GO TO 86
C     ALONG     CD
             A(NC,1) = A(NC,1) - A(NC,3)                                01297
             A(NC,2) = A(NC,2) + TWO*A(NC,3)                            01298
             A(NC,3) = ZERO                                             01299
   36    CONTINUF
      IF(I .GT. M1) GO TO 87
      IF(J .NE. LJ) GO TO 87
C     ALONG     EF
         B(NC) = B(NC) - A(NC,1)*XI(I,J)
         A(NC,1) = ZERO
   87    CONTINUE
   88 CONTINUE                                                          01300
C                                                                       01301
      N=NC
      CALL LEQT1B(A,N,NLC,NUC,IA,B,MB,IB,IJOB,XL,IER)                   01302
C                                                                       01303
      IF(IER .NE. 0) PRINT 22, IER                                      01304
   22 FORMAT('0***ERROR 1ST LEQT1B: IER= ',I5)                          01305
      NC = 0                                                            01306
         DO 96 J= 2,LM1
C     XIA(1,J) = XI(1,J)                        ***DONE UP ABOVE
      DO 89 I=2,MM1
      LJ = LAR(I)+1
         IF(J .LT. LJ) GO TO 89
         NC = NC + 1                                                    01310
         XIA(I,J) = B(NC)                                               01311
   89    CONTINUE
   96 CONTINUF                                                          01312
C
```

```
      DO 97 I=1,M                                                       01313
      LJ=LAR(I)
      XIA(I,LJ) = XI(I,LJ)
      XIA(I,L) = ZERO                                                   01315
   97 CONTINUE                                                          01316
C     NOW ALONG CD -->                                                  01317
      DO 98 J=2,LM1                                                     01318
   98 XIA(M,J) = 2*XIA(MM1,J) - XIA(MM2,J)                              01319
      IHDR = JHDR(5)                                                    01320
C  ----------------------------------------                             01322
C                                                                       01323
      IF(TS .GT. 1) GO TO 175                                           01324
C        FOR TS=1                                                       01325
         DTA = DT                                                       01326
         DO 170 I=1,M                                                   01327
         DO 170 J=1,L                                                   01328
         UA(I,J) = U(I,J)                                               01329
         VA(I,J) = V(I,J)                                               01330
  170    CONTINUE                                                       01331
      GO TO 200                                                         01332
C        FOR TS NOT 1                                                   01333
  175 CONTINUE                                                          01334
      TMP = DT/(DTA*TWO)                                                01335
         DO 180 I=1,M                                                   01336
         DO 180 J=1,L                                                   01337
         UA(I,J) = UA(I,J) +  (U(I,J)-UA(I,J)) * TMP                    01338
         VA(I,J) = VA(I,J) +  (V(I,J)-VA(I,J)) * TMP                    01339
  180    CONTINUE                                                       01340
C                                                                       01341
  200 CONTINUE                                                          01342
      NC=0                                                              01343
      DO 110 I=2,MM1                                                    01346
```

```
      EDD = EPA(I)/DY2
      LJ = LAR(I)+1
      DO 110 J=LJ,LM1
         NC = NC+1
      T = EPA(I) / TDY / Y(J)
         A(NC,3) = VA(I,J)/ TDY    - EDD   - T
         A(NC,2) = TWO/DT + (TWO*EPA(I))/DY2
         A(NC,1) = -VA(I,J)/ TDY    - EDD + T
C
         B(NC)    = EPA(I)*((XIA(I+1,J)-2*XIA(I,J)+XIA(I-1,J))/DX2
     *         - XIA(I,J)/Y(J)**2)
     *       - UA(I,J) * (XIA(I+1,J)-XIA(I-1,J))/TDX
     *         + XIA(I,J) / DT2 + VA(I,J) * XIA(I,J) / Y(J)
C
      B(NC)=B(NC)+(EPA(I+1)-EPA(I-1))/TDX *(XIA(I+1,J)-XIA(I-1,J))/TDX
     1         - (EPA(I+1)-EPA(I-1))/TDX * (
     2  (VA(I,J+1) - TWO*VA(I,J) + VA(I,J-1))/DY2  +
     3  (VA(I,J+1) - VA(I,J-1)) / TDY / Y(J)  +
     4  (VA(I+1,J) - TWO*VA(I,J) + VA(I-1,J))/DX2 - VA(I,J)/Y(J)**2)  -
     5  (EPA(I+1) - TWO*EPA(I) + EPA(I-1))/DX2    *  (
     6  (UA(I,J+1)-UA(I,J-1))/TDY + (VA(I+1,J)-VA(I-1,J))/TDX   )
C
C     ********  FOR EPS( Y )  *********
      IF(EPO .EQ. ZERO) GO TO 193
      IF( J .NE. LJ) GO TO 193
      IF( I .GT. M1) GO TO 193
         A(NC,1) = A(NC,1) + (EPA(I)-EPO)/4.0D0/DY2
         A(NC,3) = A(NC,3) - (EPA(I)-EPO)/4.0D0/DY2
      B(NC) = B(NC) + (EPA(I)-EPO)/TDY * ( (UA(I,J+1)-TWO*UA(I,J)+
     1   UA(I,J-1))/DY2 + (UA(I,J+1)-UA(I,J-1))/TWO/DY2/(LJ-1)  +
     2   (UA(I+1,J)-TWO*UA(I,J)+UA(I-1,J))/DX2 ) +
     5   (EPA(I)-TWO*EPA(I)+EPO)/DY2 *
```

```
      6   ( (UA(I,J+1)-UA(I,J-1))/TDY + (VA(I+1,J)-VA(I-1,J))/TDX )
      7   + (EPO-EPA(I))/DY/DX *
      8   ( (UA(I+1,J)-UA(I-1,J))/TDX - (VA(I,J+1)-VA(I,J-1))/TDY )
C
  193 CONTINUE
C
C     IF((I.NE.M1).AND.(I.NE.M1P1)) GO TO 102
      IF(I .NE. M1) GO TO 102
         IF(J .LE. L1) B(NC) = B(NC) + DFRODR(J)
  102 CONTINUE
      IF( J .NE. LJ) GO TO 104
         B(NC) = B(NC) - A(NC,1) * XIA(I,LJ-1)
         A(NC,1) = ZERO
  104 CONTINUE
      IF(J .NE. LM1) GO TO 106
C/*      B(NC,1) = B(NC,1) - A(NC,1) * 0.0       IE TOP WALL                 */
         A(NC,3) = ZERO
  106 CONTINUE
  110 CONTINUE
C
      N = NC
      CALL LEQT1B(A,N,NLC,NUC,IA,B,MB,IB,IJOB,XL,IER)
C
      IF(IER .NE. 0) PRINT 46,IER
   46 FORMAT('0***ERROR ***2ND LEQT1B:  IER= ',I4)
C
      NC = 0
      DO 120 I=2,MM1
      LJ = LAR(I)+1
      XIB(I,LJ-1) = XI(I,LJ-1)
         DO 120 J=LJ,LM1
         NC = NC+1
```

```
      XIB(I,J) = B(NC)
  120 CONTINUE
C
      DO 122 I=1,M
  122 XIB(I,L) = ZERO
      DO 124 J=1,L
  124 XIB(1,J) = XI(1,J)
C     NOW ALONG CD-->
      DO 126 J=2,LM1
  126 XIB(M,J) = TWO*XIB(MM1,J) - XIB(MM2,J)
      XIB(M,1) = ZERO
C
      IHDR = JHDR(6)
C
C ------------------------------------
C     DEL_SQUARE PSI  =  XI(X,Y)    A LA EHRLICH
C     FIRST SOLVE ALONG THE RHS OF THE REGION.   IE.  CD
C     BUT NOT THE END POINTS
C                                               $$$$$$$$$$$$$$$$$$$$$$$$
      DO 205 J=1,LM2
      T = DY/TWO/Y(J+1)
      A(J,3) = ONE - T
      A(J,2) = -TWO
      A(J,1) = ONE + T
      B(J) = DY2 * XIB(M,J+1) * Y(J+1)
  205 CONTINUE
C
      T = DY / TWO / Y(LM1)
      A(1,1) = ZERO
      A(LM2,2) = -ONE - T
      A(LM2,3) = ZERO
      B(LM2) = (DY2*XIB(M,LM1) - U1(M)*DY*(ONE-T)) * Y(LM1)
```

```
C                                                                      01414
      CALL LEQT1B(A,LM2,NLC,NUC,IA,B,MB,IB,IJOB,XL,IER)                01415
C                                                                      01416
      IF(IER .NE. 0) PRINT 24,IER                                      01417
   24 FORMAT('0***ERROR ***MID LEQT1B:  IER= ',I4)                     01418
      DO 130 J=1,LM2                                                   01419
      PSIB(M,J+1) = B(J)                                               01420
  130 CONTINUE                                                         01421
C                                                                      01422
C     MUST  "INTERCHANGE I AND J  FOR  POIS2  USE HERE"                01423
C     (I,J) MAPS INTO (J',I')  AND YY(30,60)                           01424
C                                                                      01425
C     ----- STEP 2 -------------
      DO 135 I=1,LM1                                                   01426
         DO 135 J=1,MM2                                                01427
      LJ = LAR(J+1)-1
      YY(I,J)=ZERO
      IF(I .GT. LJ) YY(I,J) = XIB(J+1,I+1)*DX2 * Y(I+1)
  135 CONTINUE
C
C     ALONG L.H.SIDE                                                   01430
      LJ=JCI-1
      DO 136 I=LJ,LM1
  136 YY(I,1) = YY(I,1) - PSI(1,I+1)
C     ALONG R.H.SIDE                                                   01432
      DO 2137 I=1,LM1
 2137 YY(I,MM2) = YY(I,MM2) - PSI(M,I+1)
C                                                                      01434
      DO 137 J=1,MM2                                                   01436
C     ALONG THE BOTTOM                                                 01437
C     YY(1,J) = YY(1,J) - ZERO                                         01438
C     ALONG THE TOP                                                    01439
```

```
  137 YY(LM1,J) = YY(LM1,J) - TWO*(RXY2-DX2/TWO/Y(L)/DY)*Y(L)*DY*U1(J+1)     01440
C                                                                             01441
C                                                                             01448
C     IHDR = MASK
C     CALL DMTO10(YY, LM1, MM2, L, M)
      CALL POIS2(MM2, LM1, AA,BB,CC, JDIMY, YY, XL)                           01450
C     CALL DMTO10(YY, LM1, MM2, L, M)
C                                                                             01451
      DO 210 K=2,M1M1
      LJ = LAR(K)-1
  210 PSW(K) = YY(LJ,K-1)
C     ----- STEP 3 --------------
C   4 FORMAT('0  STEP 3')
C     PRINT 4
      DO 220 K=2,M1M1
      B(K-1) = -PSW(K)
C     PRINT 99, B(K-1),K
         DO 215 I=2,M1M1
         A$(I-1,K-1) = PSWI(I,K)
  215    CONTINUE
C     PRINT 99,(A$(IP,K-1),IP=1,N$)
  220 CONTINUE
C
      CALL LEQTIF(A$,MB,N$,IA$,B,IDGT,XL,IER)
C
      IF(IER .EQ. 0) GO TO 216
      PRINT 3,IER
    3 FORMAT('0***ERROR **** LEQTIF:  IER=',I4)
  216 CONTINUE
      AI(1)=ZERO
      DO 218 I=2,M1M1
  218 AI(I) = B(I-1)
```

```
      AI(M1) = ZERO
C     ------- STEP 4 ------------
      DO 4135 J=1,MM2
      LJ = LAR(J+1)-1
         DO 4135 I=1,LM1
         YY(I,J) = ZERO
      IF(I .EQ. LJ) YY(I,J) =                    AI(J+1)
      IF(I .GT. LJ) YY(I,J) = XIB(J+1,I+1)*DX2 * Y(I+1)
 4135 CONTINUE
      LJ = JCI-1
      DO 4136 I=LJ,LM1
 4136 YY(I,1) = YY(I,1)- PSI(1,I+1)
      DO 4437 I=1,LM1
 4437 YY(I,MM2) = YY(I,MM2) - PSI(M,I+1)
      DO 4137 J=1,MM2
 4137 YY(LM1,J) = YY(LM1,J) - TWO*(RXY2-DX2/TWO/Y(L)/DY)*Y(L)*DY*U1(J+1)
C
C     IHDR = MASK
C     CALL DMTO10(YY, LM1, MM2, L, M)
      CALL POIS2(MM2, LM1, AA,BB,CC, JDIMY, YY, XL)
C     CALL DMTO10(YY, LM1, MM2, L, M)
C
      DO 138 I=1,MM2
      DO 138 J=1,LM1
  138 PSIB(I+1,J+1) = YY(J,I)
      DO 139 J=1,L
  139 PSIB(1,J) = PSI(1,J)
      DO 140 I=1,M
      PSIB(I,1) = PSI(I,1)
  140 PSIB(I,L) = PSIB(I,LM1) + U1(I)*DY* Y(L)
      IHDR = IPSIB
   30 FORMAT(' HERE IS WALL XI')
```

```
      DC 142 I=1,M1M1
      LJ = LAR(I)+1
  142 XIW(I) =TWO*(PSIB(I,LJ)/DY2 + PSIB(I+1,LJ-1)/DX2) / Y(LJ-1)
      XIW(M1) = THREE * PSIB(M1,2)/DY3
C
C    *PPINT 99, (XIW(I),I=1,M1)                                          01464
      PRINT 99, (XIW(I),I=1,M1)                                          01464
C                                                                        01465
C                                                                        01466
C     IF( UNE .EQ. 1.0D0) STOP
      RETURN
      END
```

```
      SUBROUTINE DRFAX6
C     MAIN DRIVER FOR THE AXI-SYMMETRICAL PROBLEM     JAN 1976  FAVIN
C              WITH A CORNER                       JULY 1976
      IMPLICIT REAL*8 (A-H,O-Z)                                         01001
C
C           U1 -->                                                      01003
C     J=L   +--------------------------------------+                    01004
C           |B        *                           C|                    1005
C           |         *                            |                    1006
C           |         *                            |                    1007
C   J=JDEL  .         *                            .                    1008
C   Y       .         *                            .      M A I N  6    1009
C           .         *                            .                    1010
C     J=L1  +.......*...+                          |                    1011
C           |     *     |                          |                    1012
C   J=JCI   * F         | FX                       |                    1013
C           |           |                          |                    1014
C           |           |                          |                    01015
C     J=1   |A          |E                        D|                    01016
C           +-----------+--------------------------+                    01017
C          I=1         I=M1                       I=M                   01018
C                        X                                              01019
C                                                                       01020
      REAL*8  K1, KM, KL, KU
      COMMON /REALS/  R1,C7,DX,DY,DT,U2,RHO,EPS,FX,FXRHO,X,Y,U0,
     *   DT2,DTA,TDY,TDX,DY2,  DFRODR,  TRAMP,TCL, K1, KM, KL, KU,
     *   EPO, THRUST
      COMMON /ARRAYS/  PSI,XI,V,U,VA,UA,PSIB,XIB,XIW, EPA(60), U1(60),
     *   U10(60), PSI0(60)
      COMMON /INTEGS/  L,L1,JDEL,M,M1,NTS,NER,PSKP,IDIMY,JDIMY,JCI,IO
      COMMON /IRRAYS/  LAR(60)
      COMMON  /LOGICS/  LSWH1, PRNT, ON, OFF, RSTART, SAVE, SWHFX,
```

```
      *   ZSTART, CSTART
      COMMON /PLOTZ/ SL,SU,SL1,SKEY(30)
C     DIMENSIONS ARE SET UP FOR MAXIMUMS OF  L=30, M=60, L1=15 & M1=15       01021
C                                                                            01022
      DIMENSION PSI(60,30), PSIB(60,30), XI(60,30), V(60,30), U(60,30)       01023
      DIMENSION  XIW(15),    XIB(60,30),  VA(60,30), UA(60,30)
      DIMENSION U0(30), Y(30), X(60), FX(15), FXRHO(15),  DFRODR(15)         01027
      DIMENSION DUMA(15)
      INTEGER  TS, TTS, PSKP                                                 01028
      COMMON IHDR                                                            01029
      INTEGER JHDR(6)                                                        01030
      DATA  JHDR        / 'OPSI','0 XI', '0  V', '0  U', 'OXIA', 'OXIB' /    01031
      LOGICAL  LSWH1, PRNT, ON, OFF, SWHFX
      LOGICAL  RSTART, SAVE, ZSTART, CSTART                                  01033
C                                                                            01035
      DATA ZERO,ONE,TWO,THREE,FOUR/0.0D0,1.0D0,2.0D0,3.0D0,4.0D0/            01037
C
      ON = .TRUE.
      OFF = .FALSE.
      LSWH1 = OFF
      PRINT 6                                                                01044
    6 FORMAT(' 12/30/76  CORNER AXI-SYM  **6**')
C     IA = IB =  (L-2)*(M-2)                                                 01045
C     THE DIM OF A AND B ARE N AND XL IS TWICE THAT                          01046
C     IA =  1624                                                             01047
C     IB =  1624                                                             01048
      IDIMY = 60                                                             01049
      JDIMY = 30                                                             01050
      ONE = 1.0D0
C                                                                            01038
C  READ INITIAL DATA
C
```

```
      CALL PUTIN6
C
      MM1 = M-1
      MM2 = M-2
      LM1 = L-1
      M1P1 = M1+1
C
C   SET UP INITIAL BOUNDARY CONDITIONS
C
      CALL INTAL6
C
      IF(TRAMP .EQ. ZERO) TRAMP=ONE
C                                                                       01230
C                  THE FIRST TIME STEP
C     "IT" IS ACTUAL STEP COUNTER;  IT IS ONLY UPDATED WHEN TIME IS.
      IT=0
C     FOR THIS RUN, THAT IS.
      TS = 1                                                            01232
      TTS = PSKP                                                        01233
      PRNT = ON                                                         01234
      TIME = ZERO                                                       01235
      IF( .NOT. ZSTART) GO TO 10
C     GET BETTER START FROM  **10**
      REWIND 10
      READ(10) DUM1,DUM2,DUM3,DUM4,IDUM
      READ(10) DUMA
      READ(10) PSI,XI,V,U
      CALL DU1CAL
      CALL EPSCA6( IT )
      GO TO 887
   10 IF( .NOT. RSTART)  GO TO 887                                      01236
      REWIND 9                                                          01237
```

```
      READ(9)  TIME,DT,DT2,DTA,  TS
      READ(9)  DFRCDR
      PEAD(9)  PSI,XI,V,U
      READ(9)  VA,UA
     CALL DUICAL
     CALL EPSCA6( IT )
C
C    WANT TC MAKE ANY CHANGES?
 887 IF( CSTART) CALL CHANG5( TIME, TS )
C
C   TIME_STEP LCCP:
 898 CONTINUE
     PRINT 1, TIME
   1 FORMAT('0  TIME = ',F12.5)
C    IF(TS .LE. NTS) GC TO 700                    FCR DEBUG
  13 FCRMAT('0       ******************************TIME_STEP LOCP*****
 131*** IT = ',I4)
     PRINT 13, IT
     IF(IT .EQ. 0) GC TO 402
     IF(TS .GE. NTS) GC TU 400
     GO TO 402
C
 400 CONTINUE
     IHDR = JHDR(1)
     CALL DMTO10(PSI, M, L, IDIMY, JDIMY)
     IHDR = JHDR(2)
     CALL DMTO10( XI, M, L, IDIMY, JDIMY)
     IHDR = JHDR(3)
                    CALL DMTC10(  V, M, L, IDIMY, JDIMY)
     IHDR = JHDR(4)
     CALL DMTO10(  U, M, L, IDIMY, JDIMY)
 402 CONTINUE
```

```
      PRNT = OFF
C
C  PRINT XI CONTOUR PLOT
C
      PRINT 87
   87 FORMAT('1',/////, '0          CONTOUR PLCT FOR XI')
      CALL PIXER( XI )
    4 FORMAT('1')
      PRINT 4
C                                                                  01262
  666 CONTINUE                                                     01263
      IF(TS .LT. NTS)  GC TO 700
C     SAVE FOR A LATER RESTART                                     01265
      IF( .NOT. SAVE)  GC TO 699
      PRINT 2, IO
    2 FORMAT('0  DATA SAVED ON UNIT ',I5)
      REWIND IO
      WRITE(IO)  TIME,DT,DT2,DTA,  TS
      WRITE(IO)  DFRODR
      WRITE(IO)  PSI,XI,V,U
      WRITE(IO)  VA,UA
C 699 STOP                                                         01270
  699 RETURN
  700 CONTINUE                                                     01271
      IF( .NOT. PRNT) GC TO 701
      PRINT 87
      CALL PIXER( XI )
      PRINT 4
  701 CONTINUE
C                                                                  01272
C    THE ITERATION LOOP                                            01273
C
```

```
      DO 777 ITER=1,NER                                                 01274
      PRINT 21, ITER                                                    01275
   21 FORMAT( ///,' ITER = ',I4,'     ----------------------------------')  01276
      CALL WORK6(TIME, TS, TTS )
C
C     TEST CONVERGENCE                  --------------------            01467
C
      XMAX = ZERO                                                       01468
      DO 147 I=2,M1                                                     01469
      J = JCI+1-I
      PRINT 3, I,J,XI(I,J),XIW(I)                                     DEBUG
    3 FORMAT('0I,J,XI(I,J),XIW ARE ',2I6,1P2G20.10)                   DEBUG
      IF(XI(I,J) .EQ. ZERO) GO TO 1313
      XAB = DABS((XI(I,J)-XIW(I))/XI(I,J))
      XMAX = DMAX1(XMAX,XAB)                                            01471
  147 CONTINUE                                                          01474
      PRINT 31, XMAX                                                    01475
   31 FORMAT('0  XMAX = ', 1PE12.4)                                     01476
      IF(XMAX .LT. TOL ) GO TO 555                                      01477
C
C  IF XMAX< IS GREATER THAN TOL SET XI=(XIW+XI)/2
C
         DO 146 I=2,M1
         J=JCI+1-I
         XI(I,J)  = (XI(I,J)+XIW(I))/TWO
         XIB(I,J) = XI(I,J)
  146    CONTINUE
  777 CONTINUE                                                          01478
C     END OF LOOP ON ITER---------------------------------------------  01479
C                                                                        1498
C/**** NOT KNOWING WHAT TO DO................ */                         1499
 1313 CONTINUE                                                           1500
```

```
      PRNT=ON                                                    1501
      PRINT 32                                                   1502
   32 FORMAT('OFINISHED THE LOOP AND DONT KNOW WHAT TO DO.')     1503
C         STOP BUT FIRST PRINT OUT WHAT YOU GOT                  1504
      SAVE = OFF
      TS = NTS+1                                                 1505
C     ____                                                       1506
C    OUT_OF_LOOP                                                 1507
C
  555 CONTINUE                                                   1508
      DO 556 I=2,M1
      J=JCI+1-I
      XI(I,J) = XIW(I)
      XIB(I,J) = XIW(I)
  556 CONTINUE
      TIME = TIME + DT                                           1509
      IT = IT + 1
C
      DO 35 J=1,L1
      DFRODR(J) = FXRHO(J)
      IF(TIME .LT. TRAMP) DFRODR(J) = FXRHO(J)*TIME/TRAMP
   35 CONTINUE
    9 FORMAT('0 DFRODR:')
   99 FORMAT(2X,1P10G12.4)
      PRINT 9
      PRINT 99, (DFRODR(I),I=1,L1)
C
      DTA = DT                                                   1510
      IF(ITER .GT. 3)  DT = DT/R1                                1511
      IF(ITER .LT. 3)  DT = DT*R1                                1512
  300 CONTINUE                                                   1513
      PRINT 33, ITER, TS                                         1514
```

```
   33 FORMAT('0 ITER AND TS ARE ',2I8)                              1515
      DT2 = DT / TWO                                                1516
      PRINT 34, DT                                                  1517
   34 FORMAT('0  DT IS ',F12.6)                                     1518
      TS = TS + 1                                                   1519
C
C   DEFINE THE GRID STREAM FUNCTION AND DU/DY DISTRIBUTION
C  BASED ON THE FINAL RESULTS FROM WORK6
C
      DO 150 I=1,M                                                  1520
         DO 150 J=1,L                                               1521
         XI(I,J) = XIB(I,J)                                         1522
         PSI(I,J) = PSIB(I,J)                                       1523
  150 CONTINUE                                                      1524
C
C  CALCULATE U AND V FOR THE REGION AHEAD OF THE TRAILING EDGE
C
      DO 155 I=2,M1
      JL = JCI+2-I
      DO 155 J=JL,LM1
      U(I,J) = (PSI(I,J+1) - PSI(I,J-1))/TDY/Y(J)                   1527
      V(I,J) = -(PSI(I+1,J)-PSI(I-1,J))/TDX /Y(J)                   1528
  155 CONTINUE                                                      1529
C
C  CALCULATE U AND V FOR THE WAKE REGION
C
      DO 5155 I=M1P1,MM1
      DO 5155 J=2,LM1
      U(I,J) = (PSI(I,J+1) - PSI(I,J-1))/TDY/Y(J)
      V(I,J) = -(PSI(I+1,J)-PSI(I-1,J))/TDX /Y(J)
 5155 CONTINUE
C                                                                   1530
```

```
C     RECALCULATE U(I,1) = D(PSI)/DY FUR M1P1 TO M                          1531
C
      DU 160 I=M1P1, M                                                      1532
  160 U(I,1) = TWO*PSI(I,2)/DY2                                             1533
C
C  CALCULATE U FCR THE DOWNSTREAM BCUNDARY
C
      DC 162 J=2,LM1                                                        1534
      U(M,J) = (PSI(M,J+1)-PSI(M,J-1)) / TDY/Y(J)                           1535
      V(M,J) = -(PSI(MM2,J) - FCUR*PSI(MM1,J) + THREE*PSI(M,J))/TDX/Y(J)    1536
  162 CCNTINUE                                                              1537
C
C  CALCULATE U(M,L)
C
      U(M,L) = U1(M)                                                        1538
C                                                                           1539
C  CALCULATE V(M,L)
C
      V(M,L) = -(PSI(MM2,L) - FCUR*PSI(MM1,L) + THREE*PSI(M,L))/TDX/Y(L)    1540
C
C  CALCULATE V ALCNG THE UPSTREAM BCUNDARY
C
      DC 166 I=2,MM1                                                        1541
  166 V(I,L) = -(PSI(I+1,L) - PSI(I-1,L)) / TDX /Y(L)                       1542
C                                                                           1543
      IF(MOD(IT+1,PSKP) .EQ. 0)  CALL DUICAL
      IF(MCD(IT,PSKP) .EQ. 0)  CALL EPSCA6( IT )
C
      PRNT = ON                                                             1545
      IF ( TS .GE. NTS )  GC TO 888                                         1546
      IF ( TS .LE.  2  )  GO TO 888                                         1547
      PRNT = OFF                                                            1548
```

```
      TTS = TTS - 1
      IF ( TTS .GT. 0) GO TO 888
      PRNT = ON
      TTS = PSKP
      GO TO 888
C     GO TO TIME_STEP;
      END
```

```
      SUBROUTINE PUTIN6
C
C     THIS SUBROUTINE IS RESPONSIBLE FOR READING IN THE DATA FOR
C     DRFAX6. THE BODY SHAPE IS DEFINED IN THE LAR ARRAY AND ANY
C     QUANTITIES THAT NEED BE INITIALIZED ARE SET IN THIS ROUTINE
C     AND INTAL6.
C
      IMPLICIT REAL*8 (A-H,O-Z)                                        01001
      REAL*8  K1, KM, KL, KU
      COMMON /REALS/  R1,C7,DX,DY,DT,U2,RHO,EPS,FX,FXRHO,X,Y,U0,
     *   DT2,DTA,TDY,TDX,DY2,  DFRODR,  TRAMP,TCL, K1, KM, KL, KU,
     *   EPO, THRUST
      COMMON /ARRAYS/  PSI,XI,V,U,VA,UA,PSIB,XIB,XIW, EPA(60), U1(60),
     *   U10(60), PSI0(60)
      COMMON /INTEGS/  L,L1,JDEL,M,M1,NTS,NER,PSKP,IDIMY,JDIMY,JCI,IC
      COMMON /IRRAYS/  LAR(60)
      COMMON  /LOGICS/  LSWH1, PRNT, ON, OFF, RSTART, SAVE, SWHFX,
     *   ZSTART, CSTART
      COMMON /PLOTZ/ SL,SU,SL1,SKEY(30)
C     DIMENSIONS ARE SET UP FOR MAXIMUMS OF  L=30, M=60, L1=15 & M1=15 01021
C                                                                      01022
      DIMENSION PSI(60,30), PSIB(60,30), XI(60,30), V(60,30), U(60,30) 01023
      DIMENSION  XIW(15),   XIB(60,30),  VA(60,30), UA(60,30)
      DIMENSION U0(30), Y(30), X(60), FX(15), FXRHO(15),  DFRODR(15)   01027
      INTEGER  PSKP
      COMMON IHDR                                                      01029
      LOGICAL  LSWH1, PRNT, ON, OFF, SWHFX
      LOGICAL SWH
      LOGICAL  RSTART, SAVE, ZSTART, CSTART                            01033
C                                                                      01035
      DATA ZERO,ONE                 /0.0D0,1.0D0 /                     01037
C                                                                      01038
```

```
      NAMELIST /ILIST/ L,L1,M,M1,PSKP,NER,NTS,JDEL,JCI,IO
      NAMELIST /RLIST/ DX,DY,DT,   RHO,EPS,TCL,C7,TRAMP,KU,K1,KM,KL,EPO
      NAMELIST /LLIST/ SWHFX,RSTART,SAVE,ZSTART,CSTART
C                                                                        01042
      PRINT 87
   87 FORMAT('0  PUTIN6  01/20/77   MAKE PLOT * 7',//)
C                                                                        01051
      TWO = 2.0D0/ONE
      THREE = 3.0D0
      FOUR = 4.0D0
      R1 = 1.10                                                          01053
   14 FORMAT('0**** A RESTART ****')                                     01078
   15 FORMAT('0**** SAVE FOR A PESTAPT AFTER NTS TIME_STEPS.  ***')      01079
    5 FORMAT('0***NEW START.  GET GOOD GUESS FROM UNIT 10***')
C
C     SWHFX = .TRUE. IMPLIES "NOT" TC COMPUTE A NEW FX CURVE IN SUB
C                                         INITAL
C     ZSTART=ON  IMPLIES THIS IS A NEW START AND GET A GOOD GUESS FROM
C          DATA ON UNIT 10
C     CSTART = ON IMPLIES A CALL TO SUB CHANGE TO MAKE SOME CHANGES
C          BEFORE ACTUALLY STARTING.
      TRAMP = 0.1D0
 1000 CONTINUE                                                           01080
C     INPUT PARAMETERS                                                   01081
      READ (5, ILIST)
      READ (5, RLIST)
      READ (5, LLIST)
C
      TDX = TWO*DX
      TDY = TWO*DY
C
      WRITE(6, ILIST)
```

```
      WRITE(6, RLIST)
      WRITE(6, LLIST)
C
      IF( (JCI+2 .LT. JDEL) .AND. (JCI .LE. M1) )  GO TO 20
      PRINT 4
    4 FORMAT('0***ERROR IN JCI & JDEL')
      STOP
   20 CONTINUE
C
C  DEFINING THE LAR ARRAY
C
      DO 21 I=1,M
   21 LAR(I)=1
C
      DO 22 I=1,M1
      J = JCI+1 - I
      LAR(I) = J
      IF(J.EQ.1) GO TO 23
   22 CONTINUE
   23 PRINT 6, LAR
    6 FORMAT('0THE LAR ARRY',//,(5X,20I3) )
C
      IF( RSTART ) PRINT 14
      IF( SAVE ) PRINT 15
      IF(ZSTART) PRINT 5
      PRINT 2,L,L1,M,M1,    DX,DY,DT,KU,RHO,      EPS,NER,NTS,JDEL
    2 FORMAT('0HERE IS L,L1,M,M1,    DX,DY,DT,KU,RHO,EPS,    NER,NTS,JDEL'
     2*,//,(4I6,/1P6G14.5/)  )
C
      PRINT 111, TOL
  111 FORMAT('0 TOL = ',F12.5)
      PRINT 1, TRAMP
```

```
    1 FORMAT('0 FOR GRAD PROP,  TRAMP =',F8.2)
      DO 53 I=1,L1                                                      01111
   53 FX(I) = ZERO                                                      01112
C                                                                       01113
      IF(JDEL .LE. 1) READ 3, (FX(I),I=1,L1)                            01115
    3 FORMAT(8F10.0)                                                    01116
C                                                                       01118
      SL = -7.9392829073515640D0
      SL1= -5.7328021535937510D0
      SU = 1.35D0
      SKEY(1) = SL
      SKEY(2) = SL1
      DPY = SL1 - SL
      SWH = OFF
C
      DO 35 I=3,30
      IF(SKEY(I-1)) 30,31,31
   30 DPY = DPY / SU
      GO TO 32
   31 IF(SWH) DPY = DPY * SU
      SWH = ON
   32 SKEY(I) = SKEY(I-1) + DPY
      IF( DABS(SKEY(I)) .LE. 1.0D-5) SKEY(I)=0.0D0
   35 CONTINUE
      DO 80 I=1,30
      SKEY(I) = 7.0D0 * SKEY(I)
   80 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE DUICAL
C
C     THIS SUBROUTINE CALCULATES A D(U1) WHICH IS ADDED TO THE INITIAL
C     VALUE COMPUTED IN INTAL6. THE VALUE OF D(U1) IS UPDATED DURING
C     THE ITERATION SEQUENCE TO MODIFY THE INITIAL BOUNDARY CONDITIONS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1, KM, KL, KU
      COMMON /REALS/  R1,C7,DX,DY,DT,U2,RHC,EPS,FX,FXRHC,X,Y,U0,
     * DT2,DTA,TDY,TDX,DY2,  DFRODR,  TRAMP,TOL, K1, KM, KL, KU,
     * EPO, THRUST
      COMMON /ARRAYS/  PSI,XI,V,U,VA,UA,PSIB,XIB,XIW, EPA(60), U1(60),
     * U10(60), PSI0(60)
      COMMON /INTEGS/  L,L1,JDEL,M,M1,NTS,NER,PSKP,IDIMY,JDIMY,JCI,IO
      COMMON /IRRAYS/  LAR(60)
      COMMON  /LOGICS/  LSWH1, PRNT, ON, OFF, RSTART, SAVE, SWHFX,
     *  ZSTART, CSTART
      DIMENSION PSI(60,30), PSIB(60,30), XI(60,30), V(60,30), U(60,30)      01023
      DIMENSION  XIW(15),    XIB(60,30),  VA(60,30), UA(60,30)
      DIMENSION U0(30), Y(30), X(60), FX(15), FXRHO(15),  DFRODR(15)        01027
      INTEGER  PSKP
      LOGICAL  LSWH1, PRNT, ON, OFF, SWHFX
      LOGICAL  RSTART, SAVE, ZSTART, CSTART                                 01033
C
      DIMENSION  DPSI(60), DDU1(60)
C
    1 FORMAT('    DUICAL     09/01/76')
    2 FORMAT('0    I        U1(I)          DDU1(I)          U10(I)     PSI0')
    3 FORMAT(I5,1P4G15.5)
      PRINT 1
      MM1=M-1
      RT=(L-1)*DY
```

```
      RT2 = RT**2
      ZO=0.0DO
      TWO=2.0DO
      DDU1(1)=ZO
      DPSI(M) = ZO
C
      DO 10 I=1,MM1
      DPSI(I) = PSI(I+1,L) - PSI(I,L) - PSIO(I+1) + PSIO(I)
   10 CONTINUE
C
      PRINT 2
      DO 30 I=2,M
      XT = (I-1)*DX
      DDU1(I)=ZO
       DO 20 K=1,MM1
       XK=(K-1)*DX + DX/TWO
       XTK = XT-XK
       R2 = TWO * (RT2 + XTK**2)
       BETA = DATAN2( RT, XTK )
        DV = DPSI(K) / R2
       DDU1(I) = DDU1(I) + DV*DCOS( BETA )
   20  CONTINUE
      U1(I) = U1O(I) + DDU1(I)
      PRINT 3, I, U1(I), DDU1(I), U1O(I), PSIO(I)
   30 CONTINUE
      DO 40 I=1,M
   40 U(I,L) = U1(I)
      RETURN
      END
```

```
      SUBROUTINE INTAL6
C
C     THIS SUBROUTINE SOLVES FOR THE INITIAL INVISCID FLOWFIELD
C     AND SETS THE INITIAL BOUNDARY CONDITIONS FOR THE AXISYM-
C     METRIC CASE. THE STREAM FUNCTION FOR THE INVISCID FLOWFIELD
C     IS SOLVED FOR USING A SOURCE-SINK DISTRIBUTION ALONG THE
C     BODY AXIS. THIS DISTRIBUTION IS FED INTO THE SUBROUTINE
C     FROM SUBROUTINE SINK.
C
      IMPLICIT REAL*8 (A-H,O-Z)                                         01001
      REAL*8  K1, KM, KL, KU
      COMMON /REALS/  R1,C7,DX,DY,DT,U2,RHC,EPS,FX,FXRHC,X,Y,U0,
     *   DT2,DTA,TDY,TDX,DY2,  DFRODR,  TRAMP,TCL, K1, KM, KL, KU,
     *   EPO, THRUST
      COMMON /ARRAYS/  PSI,XI,V,U,VA,UA,PSIB,XIB,XIW, EPA(60), U1(60),
     *   U10(60), PSI0(60)
      COMMON /INTEGS/  L,L1,JDEL,M,M1,NTS,NER,PSKP,IDIMY,JDIMY,JCI,IC
      COMMON /IRRAYS/  LAR(60)
      COMMON  /LOGICS/  LSWH1, PRNT, ON, OFF, RSTART, SAVE, SWHFX,
     *   ZSTART, CSTART
C     DIMENSIONS ARE SET UP FOR MAXIMUMS OF  L=30, M=60, L1=15 & M1=15  01021
C                                                                       01022
      DIMENSION PSI(60,30), PSIB(60,30), XI(60,30), V(60,30), U(60,30)  01023
      DIMENSION  XIW(15),   XIB(60,30),  VA(60,30), UA(60,30)
      DIMENSION U0(30), Y(30), X(60), FX(15), FXRHO(15),  DFRODR(15)    01027
      INTEGER  PSKP
      COMMON IHDR                                                       01029
      LOGICAL  LSWH1, PRNT, ON, OFF, SWHFX
      LOGICAL  RSTART, SAVE, ZSTART, CSTART                             01033
C                                                                       01035
      DATA ZERO,ONE,TWO,THREE,FOUR/0.0D0,1.0D0,2.0D0,3.0D0,4.0D0/       01037
C                                                                       01038
```

```
      PRINT 1
    1 FORMAT('     INTAL6     02/09/77')
      JC1 = JCI+1
      JC2 = JCI + 2
      FXO = ZERO
      PI4 = DATAN(ONE)
      TDY = TWO*DY
      LM1 = L-1
      L1M1 = L1-1
      L1M2 = L1-2
      L13 = L1-3
      L14 = L1-4
      M1P1 = M1+1
C                                                                  01114
      DO 52 I=1,M                                                  01119
   52 X(I) = (I-1)*DX                                              01120
      DO 54 I=1,L                                                  01121
      UO(I) = ZERO
      Y(I) = (I-1) * DY                                            01122
   54 CONTINUE                                                     01126
C
      DO 60 I=1,M1
      DO 60 J=1,M1
      PSI(I,J) = ZERO
       XI(I,J) = ZERO
        V(I,J) = ZERO
        U(I,J) = ZERO
   60 CONTINUE
      DO 601 I=1,M
      J1 = LAR(I)
      XP = K1 + (I-1)*DX
         DO 600 J=J1,L
```

```
      IF(J .GT. J1) GO TO 30
         V(I,J)=ZERO
         U(I,J)=ZERO
       PSI(I,J)=ZERO
      GO TO 600
   30 CONTINUE
         RB = (J-1)*DY
    2 FORMAT(1P8G15.5)
    3 FORMAT(I8,1P4G15.5)
C
C  BEGIN SOLVING FOR THE INVISCID FLOWFIELD
C
         THETA = DATAN(RB/XP)
         R = DSQRT(RB**2 + XP**2)
         CT = DCOS(THETA)
         ST2 = ONE - CT**2
         ST = DSQRT(ST2)
         RKL = R/KL
C
      DS = DSQRT(RKL**2 + ONE - TWO*RKL*CT)
C
C  CALCULATE STREAM FUNCTION (PSI)
C
         PSI(I,J) = -KM*CT + KM*(RKL - DS) + KU*R**2*ST2/TWO
         VRU = KM/KU/R**2 - KM/KU/KL**2/(RKL*DS) + CT
         VTU = -KM/KU/KL**2/(RKL*ST) * (ONE-(RKL-CT)/DS) - ST
C
         VTOT = KU * DSQRT(VRU**2 + VTU**2)
         ALPHA = DATAN(VTU/VRU)
C
         U(I,J) = VTOT * DCOS(THETA+ALPHA)
         V(I,J) = VTOT * DSIN(THETA+ALPHA)
```

```
600     CONTINUE
      U1(I) = U(I,L)
      U10(I) = U1(I)
      PSIO(I) = PSI(I,L)
  601 CONTINUE
      DC 602 I=M1P1,M
      XP = K1 + (I-1)*DX
      U(I,1) = KU*(CNE+KM/KU/KL**2*(ONE/XP**2-CNE/XP/(XP-ONE)))
  602 CUNTINUE
C
          V(M1,2) = V(M1P1,2)
          V(M1-1,3) = V(M1,3)
      DO 603 J=2,L
  603 U(M1,J) = U(M1P1,J)
C
      YBOT = Y(JDEL) - Y(JCI)
      U2 = U(1,JDEL)
      PRINT 10, Y(JDEL), YBOT, U2
   10 FCRMAT('0  Y(JDEL) = ', 1PG20.10,'   Y(JDEL)-Y(JCI)= ',1PG20.10,      01129
   10*'   U2 = ',1PG20.10)
      JDELM1 = JDEL-1                                                       01130
C
C  CALCULATE THE VELOCITY PROFILE AT THE FRONT EDGE OF THE REGION
C
      DC 56 I=JCI,JDELM1
      U0(I) = U2 * ( (Y(I)-Y(JCI))/YBOT )**C7
      U(1,I) = U0(I)
   56 CONTINUE                                                              01133
C                                                                           01134
C     SWITCH TO COMPUTE FX"S OR LET THEM BE ZERC.
C                                                                           01145
      IF(SWHFX) GC TO 59
```

```
C                                                                          01146
C     FIRST:  INTEGRAL OF (U0*(U2-U0)*Y)DY  FROM Y(JCI) TO Y(JDEL)        01147
C     SU0 = ZERO                                                           01148
C     DO 57 I=JCI,JDELM1
C  57 SU0 = SU0 + U0(I)*(U2-U0(I))*Y(I)                                    01150
C     SU0 = SU0 * DY                                                       01151
C                                                                          01152
C     FX0 =   TWO * SU0 / Y(L1)**2 / DX
C                                                                          01154
      SU0 = 0.D0
      FX0 = 259.0D0
      PRINT 12, SU0, FX0                                                   01155
   12 FORMAT('0INT OF U0*(U2-U0)*Y  FROM Y(JCI) TO Y(JDEL) AND FX0  ARE'
     12*,/1P2G20.10)
C                                                                          01158
      DO 58 I=1,L1                                                         01161
   58 FX(I) = FX0
C                                                                          01163
      DO 580 I=L13,L1
  580 FX(I) = FX0/TWO * (ONE + DCOS((Y(I)-Y(L14))*PI4/DY))
C
C                                                                          01167
   59 CONTINUE                                                             01164
      PRINT 4, (FX(I),I=1,L1)                                              01165
    4 FORMAT('0HERE IS FX(1:L1) =',/,(1P10E13.4)   )                       01166
C
C
C     PSI IS INTEGRAL (Y(JCI) TO Y(J)) OF Y*U(Y) DY (SANS 2*PI)
C           AND XI IS DU/DY
C
      HH = DY/TWO                                                          01180
      XI(1,JCI) = (-THREE*U(1,JCI)+FOUR*U(1,JC1)-U(1,JC2)) / TDY
```

```
      DO 64 J=JC1,L
      PSI(1,J) = PSI(1,J-1) + HH*(U(1,J)+U(1,J-1)) * Y(J)                  01183
      IF( J .NE. L)  XI(1,J) = (U(1,J+1)-U(1,J-1))/TDY
   64 CONTINUE                                                             01185
      XI(1,L) = ZERO
C
      IF(EPS .GT. ZERO) GC TO 65
C     THE END POINTS OF THE INTEGRAND ARE ASSUMED TO BE ZERO               01187
C     PLANAR  DSTAR  FCR EPS
C     THE FOLLOWING WAS REMOVED  2/9/77
C     DELSR2 = 0.5D0
C     DO 62 I=JC1,JDELM1
C  62 DELSR2 = DELSR2 +        ONE-U0(I)/U2                                01190
C     DELSR2 = DELSR2 * DY                                                 01191
C     EPS = 0.018D0 *      DELSR2   * U2                                   01192
C     2/9/77
      DELSR2 = (Y(JDEL)-Y(JCI)) / (CNE/C7+ONE)
      EPS = 0.0168D0 * DELSR2 * U2
      PRINT 11, DELSR2, EPS                                                01193
   11 FCRMAT('0  DELSR2 AND EPS ARE ', 1P2G17.8)                           01194
   65 CONTINUE                                                             01195
      DC 27 I=1,M
   27 EPA(I) = EPS
C     ----------
C     NOW FILL IN...                                                       01196
      K15 = L - JCI
      DO 40 I=2,M1
      LJ = LAR(I)
        DO 41 K=1,K15
        J = LJ+K-1
        XI(I,J) = XI(I-1,J+1)
   41   CONTINUE
```

```
      J1 = J+1
         DO 42 K=J1,L
         XI(I,K) = XI(I,J)
   42    CONTINUF
   40 CONTINUE
C
      DC 66 I=M1P1,M
      DO 66 J=1,L                                                        01198
         XI(I,J) = XI(M1,J)
   66 CCNTINUE                                                           01202
C     ALONG ED-->                                                        01203
      DC 78 I=M1P1,M                                                     01204
      PSI(I,1) = ZERO
   78 XI(I,1)=ZERO                                                       01205
C     ALONG THE WALL
      DO 79 I=1,M1
      J = LAR(I)
      U(I,J)=ZERO
      V(I,J)=ZERO
      PSI(I,J)=ZERO
      IF(I.EQ.1) GO TO 79
      PSI(I,J+2) = TWO*PSI(I,J+3)/THREE
      PSI(I,J+1) = PSI(I,J+3) / THREE
        U(I,J+2) = TWO*  U(I,J+3)/THREE
        U(I,J+1) =     U(I,J+3) / THREE
       V(I,J+2) = TWO*  V(I,J+3)/THREE
       V(I,J+1) =     V(I,J+3) / THREE
   79 XI(I,J) = XI(1,JCI)
C                                                                        01206
      DO 80 J=1,L1
      FXRHO(J) = ZERO
   80 DFRODR(J) = ZERO
```

```
C      NOW ONLY E    IE.  I=M1, J=2 TO L1-1                        01207
C      ASSUMING FXRHO IS ZERO INITIALLY
       DO 82 J=L13,L1
    82 FXRHO(J) = -FXO * PI4/TDY * DSIN((Y(J)-Y(L14))*PI4/DY)
C                                                                  01212
    99 FORMAT(2X,1P10G12.4)                                        01213
C                                                                  01214
     5 FORMAT('0    UO')                                           01215
       PRINT 5                                                     01216
       PRINT 99, (UO(I),I=1,L)                                     01217
     7 FORMAT('0     Y ')                                          01218
       PRINT 7                                                     01219
       PRINT 99,  (Y(I),I=1,L)                                     01220
     8 FORMAT('0     X ')                                          01221
       PRINT 8                                                     01222
       PRINT 99,  (X(I),I=1,M)                                     01223
     9 FORMAT('0    FXRHO')                                        01224
       PRINT 9                                                     01225
       PRINT 99, (FXRHO(I),I=1,L1)                                 01226
C                                                                  01227
       RETURN
C                                                                  01229
       END
```

```
      SUBROUTINE EPSCA6 ( ID )
C
C     THIS SUBROUTINE CALCULATES THE EDDY VISCOSITY LAW AS DES-
C     CRIBED IN THE ACCOMPANING REPORT. AS AN ADDITIONAL REFERENCE,
C     SEE THE SCHETZ-FAVIN PAPER CONCERNING THE DEVELOPMENT OF THE
C     PROBLEM TITLED " NUMERICAL SOLUTION FOR THE NEAR WAKE OF A
C     BODY WITH PROPELLER".
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8  K1, KM, KL, KU, I1, I2
      COMMON /REALS/  P1,C7,DX,DY,DT,U2,RHO,EPS,FX,FXRHO,X,Y,U0,
     *   DT2,DTA,TDY,TDX,DY2,  DFRODR,   TRAMP,TOL, K1, KM, KL, KU,
     *   EPO, THRUST
      COMMON /ARRAYS/  PSI,XI,V,U,VA,UA,PSIB,XIB,XIW, EPA(60), U1(60),
     *   U10(60), PSI0(60)
      INTEGER PSKP
      COMMON /INTEGS/  L,L1,JDEL,M,M1,NTS,NER,PSKP,IDIMY,JDIMY,JCI,IO
      COMMON /IRRAYS/  LAR(60)
      COMMON /LOGICS/  LSWH1, PRNT, ON, OFF, RSTART, SAVE, SWHFX,
     *   ZSTART, CSTART
C     DIMENSIONS ARE SET UP FOR MAXIMUMS OF  L=30, M=60, L1=15 & M1=15
C
      DIMENSION PSI(60,30), PSIB(60,30), XI(60,30), V(60,30), U(60,30)
      DIMENSION  XIW(15),   XIB(60,30),  VA(60,30), UA(60,30)
      DIMENSION DU(30), DUY(30), I1(60), I2(60), DI1(60), DB(60),B(60)
      DIMENSION DU1(60), RI(60)
      DIMENSION U0(30), Y(30), X(60), FX(15), FXRHO(15),  DFRODR(15)
      LOGICAL  LSWH1, PRNT, ON, OFF, SWHFX
      LOGICAL  RSTART, SAVE, ZSTART, CSTART
C
      DATA IDSV / -1 /
      IF(ID .EQ. 0) IDSV=-1
```

```
      IF(ID .EQ. IDSV) RETURN
      IF(ID .NE. 0) IDSV=ID
C
      ZO = 0.0DO
      ONE = 1.0DO
      TWO = 2.0DO
C
      EPA(1) = EPS
      PRINT 3, EPS
    3 FCRMAT(' EPS=',1PG20.10)
      CUT = 0.001DO * XI(1,JCI)
      PRINT 425, CUT
  425 FCRMAT(' CUT IS ',1PG20.8)
      IF(CUT .GT. 0.05DO) GO TO 421
      GO TO 420
  421 IF(CUT .LT. 100.0DO) GO TC 422
  420 CUT = 0.05DO
  422 CONTINUE
      PRINT 426, CUT
  426 FCRMAT(' CUT USED IS ',1PG20.8)
      PRINT 1
C   1 FCRMAT('1 EPSCA6 02/09/77      WITH A2 = 2.33 & C2=0.03',
    1 FCRMAT('1 EPSCA6 04/06/77      WITH A2 = 2.00 & C2=0.035',
     *                  /,8X,'B',11X,'I1',10X,'I2',10X,'DB',10X,
    11 'DI1',9X,'F1',10X,'F2',10X,'F3',10X,'EPN',9X,'EPA')
C
    4 FCRMAT(' I=',I6,'    XI(I,J)',/10(6X,1P10G12.4,/))
C     IF(EPS .GT. 0.0DO) RETURN                  PUT IN IF D U M M Y
      LM2 = L-2
      DO 100 I=1,M
      LI = LAR(I)
      RI(I) = DY*(LI-1)
```

```
      LI1 = LI + 1
         DO 20  J=1,LI
         DU(J) = ZO
   20    DUY(J) = ZO
         DO 99 J=LI1,LM2
         DU(J) = (U(I,J+1)-U(I,J-1)) / TDY
         DUY(J) = Y(J) * DU(J)**2
   99    CONTINUE
      IF(I .LT. M1) DU(LI) = (4.0D0*U(I,LI1)-U(I,LI+2))/TDY
      IF(I .EQ. M1) DU(LI) = 3.0D0*PSI(M1,2)/DY**3
      IF(I .LE. M1) DUY(LI) = Y(LI) * DU(LI)**2
C
    5 FORMAT(1P6G16.5)
C
C
C
         DO 10 K=2,LM2
          J = L-K
          JP = J+1
          IF(DABS(XI(I,J)) .GE. CUT)      GO TO 30
   10     CONTINUE
      PRINT 13, I
   13 FORMAT(' ***ERROR** I =',I5)
C     SOME GUESS
      JP=JDEL
      GO TO 39
C     STOP
   30 CONTINUE
      IF(JP .LE.LI)  JP=JDEL
      IF(JP .GE. LM2)  JP=JDEL
      IF(I .EQ. 1) GO TO 39
      IF(IABS(JP-LJP) .LE. 2) GO TO 39
```

```
C      OK,  BUT LET IT JUMP BY 2 ONLY
       IF(JP-LJP)  34, 39, 35
   34  JP = LJP-2
       GO TO 30
   35  JP = LJP+2
       GO TO 30
   39  CONTINUE
C
       LJP = JP
       B(I) = Y(JP)
       I1(I)=ZO
       I2(I)=ZO
       JP1 = JP-1
          DO 40 J=LI1,JP1
          I1(I) = I1(I) + U(I,J)*Y(J)
   40     I2(I) = I2(I) + DUY(J)
C
       I1(I) = I1(I) + U(I,JP)*Y(JP)/TWO
       I2(I) = I2(I) + DUY(JP)/TWO   + DUY(LI)/TWO
C
       I1(I) = DY * I1(I) / U1(I) / (B(I)-RI(I))**2
       I2(I) = DY * I2(I) / U1(I)**2
  100  CONTINUE
C
C
       MM1 = M-1
       DO 200 I=2,MM1
       DI1(I) = (DLOG(I1(I+1))-DLOG(I1(I-1))) / TDX
       DB(I)  = (DLOG( B(I+1))-DLOG( B(I-1))) / TDX
       DU1(I) = (U1(I+1)-U1(I-1)) / TDX
  200  CONTINUE
C
```

```
      DI1(1) = (DLOG(I1(2))- DLOG(I1(1))) / DX
      DB(1)  = (DLOG( B(2))- DLOG( B(1))) / DX
      DU1(1) = (U1(2)-U1(1))/DX
C
      DB(M)  = (DLOG( B(M)) - DLOG( B(M-1))) / DX
      DI1(M) = (DLOG(I1(M)) - DLOG(I1(M-1))) / DX
      DU1(M) = (U1(M)-U1(M-1))/DX
C
      C2 = 0.035D0
      A2 = 2.00D0
C
      DO 300 I=1,MM1
      RB = RI(I)/B(I)
      BA = (ONE-RB)**2
      CON1 = C2**2 * U1(I) / TWO
      CON2 = -A2*((ONE-RB**2)/BA**2) / (4.0D0*U1(I)*C2)
      F1 = CON1 * I2(I) / I1(I)
      F2 = CON2 * EPA(I)**2 / B(I)**2 / I1(I)
      F3 = EPA(I) * (DB(I)/I1(I)/TWO/BA - DI1(I)/TWO)
      F4 = -EPA(I) * DU1(I)/TWO/U1(I)
C
      EPN = EPA(I) + DX*(F1+F2+F3+F4)
C
      PRINT 2, I,B(I),I1(I),I2(I),DB(I),DI1(I),F1,F2,F3,EPN,EPA(I)
      IF(EPN .LE. ZO)  EPN = EPS
      EPA(I+1) = EPN
  300 CONTINUE
      PRINT 2, M,B(M),I1(M),I2(M),DB(M),DI1(M),F1,F2,F3,EPN,EPA(M)
    2 FORMAT(I4,1P10G12.4)
      RETURN
      END
```

```
      SUBROUTINE CHANG5(TIME, TS)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /REALS/  R1,C7,DX,DY,DT,U2,RHO,EPS,FX,FXRHC,X,Y,U0,
     *   DT2,DTA,TDY,TDX,DY2,  DFRODR,  TRAMP,TOL, K1, KM, KL, KU
      COMMON /ARRAYS/  PSI,XI,V,U,VA,UA,PSIB,XIB,XIW, EPA(60), U1(60),
     *   U10(60), PSIO(60)
      COMMON /INTEGS/  L,L1,JDEL,M,M1,NTS,NER,PSKP,IDIMY,JDIMY,JCI,IC
      COMMON /IRRAYS/  LAR(60)
      COMMON  /LOGICS/  LSWH1, PRNT, CN, OFF, RSTART, SAVE, SWHFX,
     *   ZSTART, CSTART
C     DIMENSIONS ARE SET UP FOR MAXIMUMS OF  L=30, M=60, L1=15 & M1=15      01021
C                                                                           01022
      DIMENSION PSI(60,30), PSIB(60,30), XI(60,30), V(60,30), U(60,30)      01023
      DIMENSION  XIW(15),   XIB(60,30),   VA(60,30), UA(60,30)
      DIMENSION U0(30), Y(30), X(60), FX(15), FXRHO(15),  DFRODR(15)        01027
      INTEGER  TS, PSKP
      LOGICAL  LSWH1, PRNT, CN, OFF, SWHFX
      LOGICAL  RSTART, SAVE, ZSTART, CSTART                                 01033
      COMMON /PLOTZ/ SL,SU,SL1,SKEY(30)
      PRINT 6
    6 FORMAT(' 5/13/76   DUMMY CHANG5')
      RETURN
      END
```

```
      SUBROUTINE PIXER( U )
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /PLOTZ/ SL,SU,SL1,SKEY(30)
      DIMENSION U(60,30)
      INTEGER B(15),XK, CC(8)
      INTEGER ONCE
      INTEGER ONCE
      LOGICAL*1  C(32), A(132)
      EQUIVALENCE (A(1),B(1)),  (C(1),CC(1))
C     DATA CC /'-ABC','DEFG','HIJK','LMNO','PQRS','TUVW','XYZ+','++  '/
      DATA CC /'-765','4321','0ABC','DEFG','HIJK','LMNO','PQR+','++  '/
      DATA ONCE / 1 /
C
      IF(ONCE .EQ. 0) GO TO 100
      ONCE = 0
      PRINT 35
   35 FORMAT('0      KEY TO PLOTS')
      DO 30 K=1,28
   30 PRINT 33,  SKEY(K),    C(K),  SKEY(K+1)
   33 FORMAT(5X,1PG12.3,1X,1A1,1PG12.3)
   34 FORMAT(1H1)
      PRINT 34
  100 CONTINUE
C
      DO 10 K=1,30
      J=31-K
        DO 20 I=1,60
      XK=1
      IF(U(I,J) .LT. SKEY(2))  GO TO 19
      XK=28
      IF(U(I,J) .GE. SKEY(28))  GO TO 19
C
```

```
      DO 18 L=2,28
      XK=29-L
      IF(U(I,J) .GE. SKEY(XK)) GO TO 19
   18 CONTINUE
C
        IF(XK .LE. 0) XK=1
        IF(XK .GT. 28) XK=28
   19   CALL PLTC(A, I, C(XK), I)
   20   CONTINUE
      PRINT 2, J, (A(M),M=1,60)
   10 CONTINUE
      PRINT 4
      PRINT 3
    2 FORMAT(I3,1X,60(1A1,1X))
    4 FORMAT(4X,'|',17X'|',5(19X,'|'))
    3 FORMAT(4X,'1',17X,'10',18X,'20',18X,'30',18X,'40',18X, '50',18X,
     3*'60',//)
      RETURN
      END
```

```
      SUBROUTINE POIS2(N,M,A,B,C,IDIMY,Y,W)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8    Y(IDIMY,1),A(1),B(1),C(1),W(1)
C
C     THIS SUBROUTINE SOLVES THE LINEAR SYSTEM OF EQUATIONS
C
C        A(I)*X(I-1,J) + B(I)*X(I,J) + C(I)*X(I+1,J) +
C             X(I,J-1) - 2*X(I,J) + X(I,J+1)  =   Y(I,J) ,
C
C                   FOR I = 1,2, . . . , M , AND
C                       J = 1,2, . . . , N ,
C
C     WHERE
C
C        X(I,0) = X(I,N+1) = 0  FOR ALL I, AND
C        X(0,J) = X(M+1,J) = 0  FOR ALL J.
C
C * * * * * * * * * * * * *   RESTRICTIONS     * * * * * * * * * * * * ** *
C
C     M AND N MUST BE GREATER THAN 1.
C     W MUST BE DIMENSIONED AT LEAST 2*N + 4*M + M*(LOG2(N)+1).
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * **
C
C     PI (MACHINE DEPENDENT CONSTANT) DEFINED IN POSGN2 AND COSGEN.
C
      I1 = 2*N + 1
      I2 = I1 + M
      I3 = I2 + M
      I4 = I3 + M
      I5 = I4 + M
      DO 100 I=1,M
```

```
100 W(I4+I-1) = B(I) - 2.D0
    CALL POSGN2(N,M,A,W(I4),C,IDIMY,Y,W(1),W(I1),W(I2),W(I3),W(I5))
    RETURN
    END
```

```
      SUBROUTINE POSGN2(N,M,BA,BB,BC,IDIMQ,Q,TCOS,B,D,W,P)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8    Q(IDIMQ,1),BA(1),BB(1),BC(1),TCOS(1),B(1),D(1),W(1),P(1)
      PI = 3.14159265358979D0
      IDEGBR = 0
      IDEGCR = 0
      IP = 0
      DO 100 I=1,M
  100 P(I) = 0.D0
      NUN = N
      JST = 1
      JSP = N
      IRREG = 1
C     IRREG = 1 WHEN NO IRREGULARITIES HAVE OCCURRED, OTHERWISE IT IS 2.
  200 L = 2*JST
      NODD = 2 - 2*((NUN+1)/2) + NUN
C     NODD = 1 WHEN NUN IS ODD, OTHERWISE IT IS 2.
      GO TO (220,210),NODD
  210 JSP = JSP - L
      GO TO 230
  220 JSP = JSP - JST
      IF (IRREG .NE. 1) JSP = JSP - L
  230 CONTINUE
C     REGULAR REDUCTION
      DO 300 I=1,JST
  300 TCOS(I) = 2.D0*DCOS(DFLOAT(2*I-1)*PI/DFLOAT(L))
      IF (L .GT. JSP)  GO TO 400
      DO 380 J=L,JSP,L
      IF (JST .NE. 1)  GO TO 330
      DO 310 I=1,M
      B(I) =2.D0*Q(I,J)
  310 Q(I,J) = Q(I,J-1) + Q(I,J+1)
```

```
      CALL TRIX(JST,0,M,BA,BB,BC,B,TCOS,D,W)
      DO 320 I=1,M
  320 Q(I,J) = Q(I,J) + B(I)
      GO TO 380
  330 DO 340 I=1,M
      T = Q(I,J) - Q(I,J-JSH) - Q(I,J+JSH) + Q(I,J-JST) + Q(I,J+JST)
      B(I) = T + Q(I,J) - Q(I,J-3*JSH) - Q(I,J+3*JSH)
  340 Q(I,J) = T
      CALL TRIX(JST,0,M,BA,BB,BC,B,TCOS,D,W)
      DO 350 I=1,M
  350 Q(I,J) = Q(I,J) + B(I)
  380 CONTINUE
C     REDUCTION FOR LAST UNKNOWN
  400 GO TO (410,470),NODD
  410 GO TO (540,420), IRREG
  420 JSP = JSP + L
      J = JSP
  429 DO 430 I=1,M
  430 B(I)=.5D0*(Q(I,J-JSH)+Q(I,J-3*JSH)-Q(I,J-JST)) +P(IP*M+ I)-Q(I,J)
  432 CALL TRIX(JST,0,M,BA,BB,BC,B,TCOS,D,W)
      IP = IP + 1
      DO 431 I=1,M
      P(IP*M+I ) = .5D0*(Q(I,J-JSH)+Q(I,J+JSH)-Q(I,J)) + B(I)
  431 B(I) = Q(I,J+JST) - P(IP*M+I)
      CALL COSGEN(JSTSAV,IDEGBR,IDEGCR,TCOS)
      KK = JSTSAV*(IDEGBR+IDEGCR)
      DO 440 I=1,JST
  440 TCOS(KK+I) = 2.D0*DCOS(DFLOAT(2*I-1)*PI/DFLOAT(L))
      CALL TRIX(IDEGBR*JSTSAV,IDEGCR*JSTSAV+JST,M,BA,BB,BC,B,TCOS,D,W)
      DO 460 I=1,M
  460 Q(I,J) = Q(I,J-JST)  + B(I) - P(IP*M+I)
      IDEGCR = IDEGBR
```

```
      IDEGBR = IDEGBR + 2*JST/JSTSAV
      GO TO 540
  470 JSP = JSP + L
      J = JSP
      IP = IP + 1
      GO TO (480,500),IRREG
  480 IRREG = 2
      JSTSAV = JST
      IDEG = JST
      IDEGBR = 2
      IF (JST .NE. 1)  GO TO 483
      DO 482 I=1,M
  482 B(I) = Q(I,J)
      GO TO 528
  483 DO 485 I=1,M
      B(I) = Q(I,J) + .5D0*(Q(I,J-JST)-Q(I,J-JSH)-Q(I,J-3*JSH))
  485 P(M*(IP-1)+I) = .5D0*(C(I,J-JSH)+Q(I,J+JSH)-Q(I,J))
      GO TO 528
  500 CALL COSGEN(JSTSAV,IDEGBR,IDEGCR,TCOS)
      IDEG = IDEGBR*JSTSAV
      IDEGBR = IDEGBR + JST/JSTSAV
      DO 505 I=1,M
  505 B(I) = Q(I,J) + .5D0*(Q(I,J-JST)-Q(I,J-JSH)-Q(I,J-3*JSH))
  528 CALL TRIX(IDEG,IDEGCR*JSTSAV,M,BA,BB,BC,B,TCOS,D,W)
      DO 530 I=1,M
      P(M*IP+I) = P(M*(IP-1) + I) - B(I)
  530 Q(I,J) = Q(I,J-JST) - P(M*IP+I)
  540 NUN = NUN/2
      JSH = JST
      JST = 2*JST
  550 IF (NUN .GE. 2)  GO TO 200
      J = JSP
```

```
    DO 610 I=1,M
610 B(I) = Q(I,J)
    GO TO (620,630),IRREG
620 DO 625 I=1,JST
625 TCOS(I) = 2.D0*DCOS(DFLOAT(2*I-1)*PI/DFLOAT(2*JST))
    IDEG = JST
    GO TO 634
630 IDEGBR = IDEGCR + JST/JSTSAV
    CALL COSGEN(JSTSAV,IDEGBR,IDEGCR,TCOS)
    IDEG = JSTSAV*IDEGBR
634 CONTINUE
    CALL TRIX(IDEG,IDEGCR*JSTSAV,M,BA,BB,BC,B,TCOS,D,W)
    GO TO (635,637),IRREG
635 DO 636 I=1,M
636 Q(I,J) = .5D0*(Q(I,J-JSH)+Q(I,J+JSH)-Q(I,J)) - B(I)
    GO TO 655
637 DO 650 I=1,M
650 Q(I,J) = P(IP*M+I) - B(I)
    IP = IP - 1
655 CONTINUE
    JST = JST/2
    JSH = JST/2
    NUN = 2*NUN
    IF (NUN .GT. N)  GO TO 760
    DO 750 J=JST,N,L
    IF (J .GT. JST) GO TO 670
    DO 660 I=1,M
660 B(I) = Q(I,J) - Q(I,J+JST)
    GO TO 710
670 IF (J+JST .LE. N)GO TO 690
    DO 680 I=1,M
680 B(I) = Q(I,J) - Q(I,J-JST)
```

```
      IF (JST .LT. JSTSAV) IRREG = 1
      GO TO (710,720), IRREG
  690 DO 700 I=1,M
  700 B(I) = Q(I,J) - Q(I,J-JST) - Q(I,J+JST)
  710 DO 715 I=1,JST
  715 TCOS(I) = 2.DO*DCOS(DFLOAT(2*I-1)*PI/DFLOAT(L))
      IDEG = JST
      JDEG = 0
      GO TO 721
  720 IF (J+L .GT. N)  IDEGCR = IDEGCR - JST/JSTSAV
      IDEGBR = JST/JSTSAV + IDEGCR
      CALL COSGEN(JSTSAV,IDEGBR,IDEGCR,TCOS)
      IDEG = IDEGBR*JSTSAV
      JDEG = IDEGCR*JSTSAV
  721 CONTINUE
      CALL TRIX(IDEG,JDEG          ,M,BA,BB,BC,B,TCOS,D,W)
      IF (JST .GT. 1)  GO TO 723
      DO 722 I=1,M
  722 Q(I,J) = -B(I)
      GO TO 750
  723 IF (J+JST .GT. N)  GO TO 730
  725 DO 724 I=1,M
  724 Q(I,J) = .5DO*(Q(I,J-JSH)+Q(I,J+JSH)-Q(I,J)) - B(I)
      GO TO 750
  730 GO TO (725,739),IRREG
  739 DO 740 I=1,M
  740 Q(I,J) = P(IP*M+I) - B(I)
  750 CONTINUE
      IP = IP - 1
  755 L = L/2
      GO TO 655
  760 RETURN
      END
```

```
      SUBROUTINE TRIX(IDEGBR,IDEGCR,M,A,B,C,Y,TCOS,D,W)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8    A(1),B(1),C(1),Y(1),T  COS(1),D(1),W(1)
C
C     SUBROUTINE TO SOLVE TRIDIAGONAL SYSTEMS
C
      MM1 = M - 1
      DELTC = DFLOAT(IDEGBR+1)/DFLOAT(IDEGCR+1)
      L = DELTC
      LINT = 1
      DO 300 K=1,IDEGBR
      X =   TCOS(K)
      IF (K .NE. L)  GC TC 90
      XX = TCOS(IDEGBR + LINT) - X
      DO 20 I=1,M
      W(I) = Y(I)
   20 Y(I) = XX*Y(I)
   90 CONTINUE
      D(1) = C(1)/(X - B(1))
      Y(1) = Y(1)/(X - B(1))
      IM1 = 1
      DO 100 I=2,M
      Z = X - B(I) - A(I) *D(I-1)
      D(I) = C(I)/Z
  100 Y(I) = (Y(I) + A(I)*Y(I-1))/Z
      DC 200 IP=1,MM1
      I = M - IP
  200 Y(I) = Y(I) + D(I)*Y(I+1)
      IF (K .NE. L)  GO TC 300
      DO 250 I=1,M
  250 Y(I) = Y(I) + W(I)
      LINT = LINT + 1
```

```
C       L = (LINT*(IDEGBR+1.))/(IDEGCR+1.)
        L = DFLOAT(LINT*(IDEGBR+1))/DFLOAT(IDEGCR+1)
  300 CONTINUE
        RETURN
        END
```

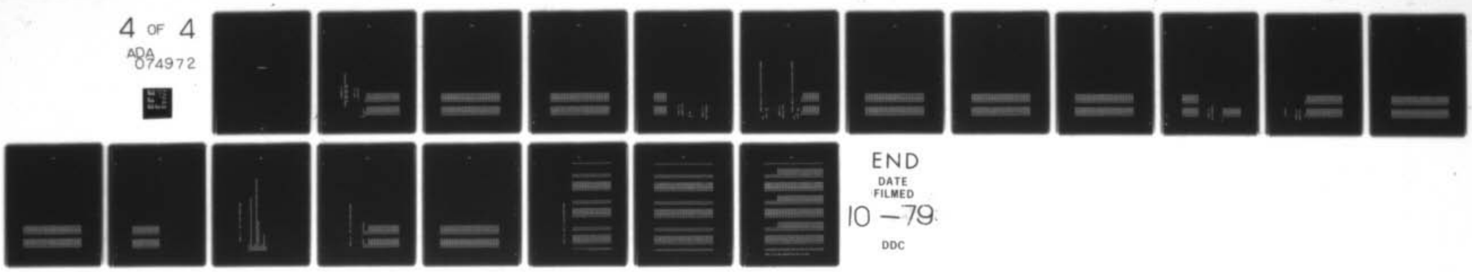
```
      SUBROUTINE COSGEN(N,M1,M2,TCOS)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8    TCOS(1)
      PI = 3.14159265358979D0
      KK = 0
      DO 200 K=1,N
      DO 100 I=1,M1
       X = DFLOAT(K) - DFLOAT(I)/DFLOAT(M1+1)
  100 TCOS(KK+I) = 2.D0*DCOS(X*PI/DFLOAT(N))
  200 KK = KK + M1
      IF (M2 .EQ. 0)  GO TO 500
      DO 400 K=1,N
      DO 300 I=1,M2
       X = DFLOAT(K) - DFLOAT(I)/DFLOAT(M2+1)
  300 TCOS(KK+I) = 2.D0*DCOS(X*PI/DFLOAT(N))
  400 KK = KK + M2
  500 RETURN
      END
```

```
      SUBROUTINE DMTO10(A,NR,NC,NRDIM,NJDIM)
      COMMON IHDR
    5 FORMAT(1A4)
C     PRINT 10 COLUMNS PER ROW WITH G12.4 FORMAT
C
C     THIS FORTRAN IV SUBROUTINE WRITES A DOUBLE PRECISION MATRIX IN A
C     READABLE FORMAT WITH ROW AND COLUMN IDENTIFICATION.
C
C     THE ARGUMENTS IN THE CALLING STATEMENT ARE:
C    A        THE MATRIX TO BE WRITTEN (SPECIFIED DOUBLE PRECISION),
C    NR       THE NUMBER OF ROWS OF A TO BE WRITTEN,
C    NC       THE NUMBER OF COLUMNS OF A TO BE WRITTEN,
C    NRDIM    THE NUMBER OF ROWS OF A IN THE DIMENSION STATEMENT OF THE
C             CALLING PROGRAM.
C
      DOUBLE PRECISION A(NRDIM,NJDIM)
      NCD8 = NC/10
      NCRM = MOD(NC,10)
      IF (NCD8 .LE. 0) GO TO 2
      DO 1 I=1,NCD8
      PRINT 5, IHDR
      I8 = I*10
      I8M7 = I8-9
      WRITE(6,100)(JCOL,JCOL=I8M7,I8)
  100 FORMAT(4H ROW,3X,4HCOL ,I3, 9(5X,4HCOL ,I3))
      DO 1 IROW=1,NR
      WRITE(6,101) IROW,(A(IROW,JCOL),JCOL=I8M7,I8)
  101 FORMAT(1H ,I3,1P10G12.4)
    1 CONTINUE
    2 IF (NCRM .LE. 0) GO TO 4
      PRINT 5, IHDR
      NCMNCR = NC-NCRM+1
```

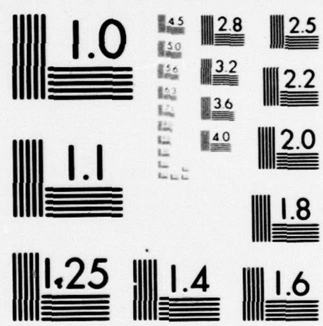
```
      WRITE(6,100) (JCOL,JCOL=NCMNCR,NC)
      DO 3 IROW=1,NR
      WRITE(6,101) IROW,(A(IROW,JCOL),JCOL=NCMNCR,NC)
    3 CONTINUE
    4 RETURN
      END
```

AD-A074 972 VIRGINIA POLYTECHNIC INST AND STATE UNIV BLACKSBURG --ETC F/G 20/4
USER'S GUIDE FOR A SYSTEM OF COMPUTER PROGRAMS TO PREDICT FLOWS--ETC(U)
NOV 78 M C HYMAN, S C HOULIHAN, J A HILL
UNCLASSIFIED VPI-AERO-096 NL

4 OF 4
ADA
074972



END
DATE
FILMED
10 -79
DDC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU

APPENDIX B

APPENDIX B

SAMPLE DATA SETS FOR EXECUTION OF TEST CASE TWO

TABLE 1.1

COORD DATA

```
0
90 15115
   0.0          0.0
   0.006667     0.023329
   0.013333     0.032712
   0.020000     0.039717
   0.026667     0.045457
   0.033333     0.050366
   0.040000     0.054670
   0.046667     0.058501
   0.053333     0.061947
   0.060000     0.065070
   0.066667     0.067914
   0.073333     0.070513
   0.080000     0.072893
   0.086667     0.075075
   0.093333     0.077075
   0.100000     0.078909
   0.106667     0.080586
   0.113333     0.082117
   0.120000     0.083509
```

0.126667	0.084770
0.133333	0.085905
0.140000	0.086919
0.146667	0.087816
0.153333	0.088601
0.160000	0.089275
0.166667	0.089841
0.173333	0.090302
0.180000	0.090659
0.186666	0.090913
0.193333	0.091065
0.200000	0.091116
0.240747	0.091116
0.281494	0.091116
0.322241	0.091116
0.362988	0.091116
0.403735	0.091116
0.444482	0.091116
0.485230	0.091116
0.525977	0.091116
0.566724	0.091116
0.607471	0.091116
0.614013	0.091115
0.620555	0.091106
0.627097	0.091084
0.633639	0.091042
0.640182	0.090974
0.646724	0.090874
0.653266	0.090738
0.659808	0.090560
0.666350	0.090337
0.672892	0.090063

0.679434	0.089736
0.685977	0.089352
0.692519	0.088909
0.699061	0.088402
0.705603	0.087831
0.712145	0.087192
0.718687	0.086484
0.725230	0.085706
0.731772	0.084856
0.738314	0.083932
0.744856	0.082934
0.751398	0.081862
0.757940	0.080715
0.764482	0.079492
0.771025	0.078193
0.777567	0.076820
0.780000	0.076281
0.790000	0.074000
0.800000	0.070583
0.810000	0.064683
0.820000	0.059265
0.830000	0.054290
0.840000	0.049722
0.850000	0.045528
0.860000	0.041677
0.870000	0.038140
0.880000	0.034893
0.890000	0.031912
0.900000	0.029174
0.910000	0.026661
0.920000	0.024352
0.930000	0.022233

0.940000	0.020287
0.950000	0.018500
0.960000	0.016859
0.970000	0.015353
0.980000	0.013969
0.990000	0.012699
1.000000	0.011533

TABLE 1.2

SEPARA DATA

```
 1
71 90
0
```

TABLE 1.3

HESS PARMDATA

```
        NSRDC SUB(L/D = 10.975) WITH ELLIPTIC FOREBODY
101

 1001  101
         1


  TABLE 2.4

 HESSBLI DATA


        NSRDC SUB(L/D = 10.975) WITH ELLIPTIC FOREBODY
101
    0
 1001  100
         1
    0.0          0.0
    0.006667     0.023329
    0.013333     0.032712
    0.020000     0.039717
    0.026667     0.045457
    0.033333     0.050366
    0.040000     0.054670
    0.046667     0.058501
    0.053333     0.061947
    0.060000     0.065070
```

0.066667	0.067914
0.073333	0.070513
0.080000	0.072893
0.086667	0.075075
0.093333	0.077075
0.100000	0.078909
0.106667	0.080586
0.113333	0.082117
0.120000	0.083509
0.126667	0.084770
0.133333	0.085905
0.140000	0.086919
0.146667	0.087816
0.153333	0.088601
0.160000	0.089275
0.166667	0.089841
0.173333	0.090302
0.180000	0.090659
0.186666	0.090913
0.193333	0.091065
0.200000	0.091116
0.240747	0.091116
0.281494	0.091116
0.322241	0.091116
0.362988	0.091116
0.403735	0.091116
0.444482	0.091116
0.485230	0.091116
0.525977	0.091116
0.566724	0.091116
0.607471	0.091116
0.614013	0.091115

0.620555	0.091106
0.627097	0.091084
0.633639	0.091042
0.640182	0.090974
0.646724	0.090874
0.653266	0.090738
0.659808	0.090560
0.666350	0.090337
0.672892	0.090063
0.679434	0.089736
0.685977	0.089352
0.692519	0.088909
0.699061	0.088402
0.705603	0.087831
0.712145	0.087192
0.718687	0.086484
0.725230	0.085706
0.731772	0.084856
0.738314	0.083932
0.744856	0.082934
0.751398	0.081862
0.757940	0.080715
0.764482	0.079492
0.771025	0.078193
0.777567	0.076820
0.780000	0.075371
0.790000	0.074053
0.800000	0.071857
0.810000	0.069693
0.820000	0.067561
0.830000	0.065459
0.840000	0.063389

0.850000	0.061348
0.860000	0.059336
0.870000	0.057354
0.880000	0.055401
0.890000	0.053476
0.900000	0.051579
0.910000	0.049710
0.920000	0.047867
0.930000	0.046052
0.940000	0.044263
0.950000	0.042500
0.960000	0.040762
0.970000	0.039050
0.980000	0.037362
0.990000	0.035699
1.000000	0.034060
1.009999	0.032445
1.019999	0.030853
1.029998	0.029285
1.039997	0.027739
1.049996	0.026216
1.059996	0.024715
1.069995	0.023235
1.079994	0.021777
1.089993	0.020341
1.099993	0.018925
1.109992	0.017529
1.119991	0.016154
1.129991	0.014799
1.139990	0.013464
1.149989	0.012148
1.159988	0.010851

1.169988	0.009573
1.179987	0.008313
1.189986	0.007072
1.199986	0.005849
1.209985	0.004643
1.219984	0.003455
1.229983	0.002285
1.239983	0.001131
1.249982	-0.000006

TABLE 2.5

WAKE COORDATA

11

1.020000
1.050000
1.100000
1.200000
1.300000
1.400000
1.500000
1.600000
1.700000
1.800000

1.900000

TABLE 2.6

SUMDS DATA

0.0	0.0
0.006667	0.024474
0.013333	0.036000
0.020000	0.045676
0.026667	0.054476
0.033333	0.062757
0.040000	0.070693
0.046667	0.078383
0.053333	0.085888
0.060000	0.093250
0.066667	0.100499
0.073333	0.107654
0.080000	0.114733
0.086667	0.121748
0.093333	0.128708
0.100000	0.135623
0.106667	0.142497
0.113333	0.149337
0.120000	0.156148
0.126667	0.162933
0.133333	0.169695

0.140000	0.176439
0.146667	0.183166
0.153333	0.189878
0.160000	0.196579
0.166667	0.203270
0.173333	0.209952
0.180000	0.216629
0.186666	0.223300
0.193333	0.229968
0.200000	0.236636
0.240747	0.277383
0.281494	0.318129
0.322241	0.358876
0.362988	0.399623
0.403735	0.440370
0.444482	0.481117
0.485230	0.521865
0.525977	0.562612
0.566724	0.603359
0.607471	0.644106
0.614013	0.650648
0.620555	0.657190
0.627097	0.663732
0.633639	0.670274
0.640182	0.676817
0.646724	0.683360
0.653266	0.689904
0.659808	0.696448
0.666350	0.702994
0.672892	0.709541
0.679434	0.716092
0.685977	0.722646

0.692519	C.729203
0.699061	0.735764
0.705603	0.742331
0.712145	0.748904
0.718687	0.755485
0.725230	0.762074
0.731772	0.768671
0.738314	0.775278
0.744856	0.781895
0.751398	0.788525
0.757940	0.795166
0.764482	0.801822
0.771025	0.808492
0.777567	0.815178
0.780000	0.818010
0.790000	0.828096
0.800000	0.838335
0.810000	C.848566
0.820000	0.858791
0.830000	0.869009
0.840000	0.879221
0.850000	0.889427
0.860000	0.899628
0.870000	0.909822
0.880000	0.920011
0.890000	0.930195
0.900000	0.940373
0.910000	0.950546
0.920000	0.960715
0.930000	0.970878
0.940000	0.981037
0.950000	0.991191

0.960000	1.001341
0.970000	1.011486
0.980000	1.021627
0.990000	1.031764
1.000000	1.041897
1.009998	1.052024
1.019999	1.062150
1.029998	1.072271
1.039996	1.082388
1.049995	1.092502
1.059996	1.102613
1.069995	1.112721
1.079993	1.122825
1.089993	1.132926
1.099993	1.143025

TABLE 3.1 UNIT 5 DATA (APLNS DATA)

```
&ILIST
NTS=70,PSKP=5,NER=40,L=30,L1=12,M=60,M1=15,IO=9,
&END
&RLIST
DT=0.1D-02,RHO=0.7535D-01,EPS=0.0,TOL=0.02D0,TRAMP=.1D0,EPG=0.D0,
SCALE1=10.0D0,SCALE2=1.0D0,
&END
&LLIST
SWHFX=F,SAVE=F,
&END
```

TABLE 3.2 UNIT 8 DATA (APLNS VELDATA)

0.100000	0.006000101
0.0	0.0
0.0	0.0
0.000000	0.000091
0.000000	0.000190
0.000000	0.000297
0.000000	0.000415
0.000000	0.000543
0.000000	0.000683
0.000001	0.000835
0.000001	0.001001
0.000001	0.001181
0.000001	0.001378
0.000001	0.001593
0.000001	0.001827
0.000001	0.002082
0.000002	0.002360
0.000002	0.002663
0.000002	0.002993
0.000002	0.003352

0.000003	0.003744
0.000003	0.004171
0.000003	0.004637
0.000004	0.005144
0.000004	0.005697
0.000005	0.006299
0.000005	0.006955
0.000005	0.007670
0.000006	0.008449
0.000007	0.009297
0.000007	0.010222
0.000008	0.011229
0.000009	0.012326
0.000010	0.013521
0.000011	0.014822
0.000012	0.016240
0.000013	0.017784
0.000014	0.019465
0.000015	0.021295
0.000017	0.023288
0.000018	0.025458
0.000020	0.027819
0.000022	0.030390
0.000024	0.033187
0.000026	0.036230
0.000029	0.039541
0.000031	0.043143
0.000034	0.047059
0.000037	0.051318
0.000041	0.055947
0.000044	0.060977
0.000048	0.066442

TABLE 3.3 UNIT 11 DATA (STREAM DATA)

1	0.903550	0.0	0.977568	0.0	0.977568	0.0	0
2	0.909980	0.0	0.975612	0.0	0.975612	0.0	0
3	0.916410	0.0	0.973802	0.0	0.973802	0.0	0
4	0.922840	0.0	0.972125	0.0	0.972125	0.0	0
5	0.929270	0.0	0.970569	0.0	0.970569	0.0	0
6	0.935700	0.0	0.969122	0.0	0.969122	0.0	0
7	0.942130	0.0	0.967772	0.0	0.967772	0.0	0
8	0.948560	0.0	0.966509	0.0	0.966509	0.0	0
9	0.954990	0.0	0.965324	0.0	0.965324	0.0	0
10	0.961420	0.0	0.964206	0.0	0.964206	0.0	0
11	0.967850	0.0	0.963145	0.0	0.963145	0.0	0
12	0.974280	0.0	0.962134	0.0	0.962134	0.0	0
13	0.980710	0.0	0.961163	0.0	0.961163	0.0	0
14	0.987140	0.0	0.960222	0.0	0.960222	0.0	0
15	0.993570	0.0	0.959300	0.0	0.959300	0.0	0
16	1.000000	0.0	0.958387	0.0	0.958387	0.0	0
17	1.006429	0.0	0.957469	0.0	0.957469	0.0	0
18	1.012857	0.0	0.956531	0.0	0.956531	0.0	0
19	1.019287	0.0	0.955554	0.0	0.955554	0.0	0
20	1.025718	0.0	0.954512	0.0	0.954512	0.0	0
21	1.032146	0.0	0.953369	0.0	0.953369	0.0	0

22	1.038576	0.0	0.952079	0.0	0.952079	0.0	0
23	1.045007	0.0	0.950572	0.0	0.950572	0.0	0
24	1.051435	0.0	0.948750	0.0	0.948750	0.0	0
25	1.057865	0.0	0.946452	0.0	0.946452	0.0	0
26	1.064296	0.0	0.943439	0.0	0.943439	0.0	0
27	1.070724	0.0	0.939352	0.0	0.939352	0.0	0
28	1.077154	0.0	0.933695	0.0	0.933695	0.0	0
29	1.083585	0.0	0.925982	0.0	0.925982	0.0	0
30	1.090014	0.0	0.916524	0.0	0.916524	0.0	0
31	1.096443	0.0	0.908230	0.0	0.908230	0.0	0
32	1.102874	0.0	0.906467	0.0	0.906467	0.0	0
33	1.109303	0.0	0.912748	0.0	0.912748	0.0	0
34	1.115732	0.0	0.922815	0.0	0.922815	0.0	0
35	1.122163	0.0	0.932782	0.0	0.932782	0.0	0
36	1.128592	0.0	0.941273	0.0	0.941273	0.0	0
37	1.135021	0.0	0.948203	0.0	0.948203	0.0	0
38	1.141452	0.0	0.953836	0.0	0.953836	0.0	0
39	1.147882	0.0	0.958454	0.0	0.958454	0.0	0
40	1.154310	0.0	0.962290	0.0	0.962290	0.0	0
41	1.160741	0.0	0.965524	0.0	0.965524	0.0	0
42	1.167171	0.0	0.968282	0.0	0.968282	0.0	0
43	1.173599	0.0	0.970663	0.0	0.970663	0.0	0
44	1.180030	0.0	0.972740	0.0	0.972740	0.0	0
45	1.186460	0.0	0.974568	0.0	0.974568	0.0	0
46	1.192888	0.0	0.976189	0.0	0.976189	0.0	0
47	1.199319	0.0	0.977638	0.0	0.977638	0.0	0
48	1.205749	0.0	0.978940	0.0	0.978940	0.0	0
49	1.212177	0.0	0.980117	0.0	0.980117	0.0	0
50	1.218608	0.0	0.981186	0.0	0.981186	0.0	0
51	1.225038	0.0	0.982162	0.0	0.982162	0.0	0
52	1.231466	0.0	0.983056	0.0	0.983056	0.0	0
53	1.237897	0.0	0.983878	0.0	0.983878	0.0	0

54	1.244327	0.0	0.984637	0.0	0.984637	0.0	0
55	1.250755	0.0	0.985339	0.0	0.985339	0.0	0
56	1.257186	0.0	0.985990	0.0	0.985990	0.0	0
57	1.263616	0.0	0.986596	0.0	0.986596	0.0	0
58	1.270044	0.0	0.987161	0.0	0.987161	0.0	0
59	1.276475	0.0	0.987690	0.0	0.987690	0.0	0
60	1.282905	0.0	0.988184	0.0	0.988184	0.0	0
61	0.903550	0.001670	0.977565	0.000264	0.977566	0.015468	0
62	0.909980	0.001670	0.975610	0.000245	0.975610	0.014372	0
63	0.916410	0.001670	0.973800	0.000227	0.973800	0.013349	0
64	0.922840	0.001670	0.972123	0.000211	0.972123	0.012409	0
65	0.929270	0.001670	0.970567	0.000196	0.970567	0.011550	0
66	0.935700	0.001670	0.969120	0.000182	0.969120	0.010762	0
67	0.942130	0.001670	0.967770	0.000170	0.967770	0.010047	0
68	0.948560	0.001670	0.966508	0.000159	0.966508	0.009437	0
69	0.954990	0.001670	0.965323	0.000150	0.965323	0.008897	0
70	0.961420	0.001670	0.964204	0.000142	0.964204	0.008412	0
71	0.967850	0.001670	0.963145	0.000135	0.963145	0.008032	0
72	0.974280	0.001670	0.962134	0.000129	0.962134	0.007705	0
73	0.980710	0.001670	0.961162	0.000125	0.961162	0.007444	0
74	0.987140	0.001670	0.960221	0.000121	0.960221	0.007241	0
75	0.993570	0.001670	0.959300	0.000120	0.959300	0.007148	0
76	1.000000	0.001670	0.958387	0.000119	0.958387	0.007121	0
77	1.006429	0.001670	0.957469	0.000121	0.957469	0.007219	0
78	1.012857	0.001670	0.956532	0.000124	0.956532	0.007454	0
79	1.019287	0.001670	0.955555	0.000131	0.955555	0.007869	0
80	1.025718	0.001670	0.954513	0.000142	0.954513	0.008506	0
81	1.032146	0.001670	0.953372	0.000157	0.953372	0.009443	0
82	1.038576	0.001670	0.952082	0.000180	0.952082	0.010834	0
83	1.045007	0.001670	0.950578	0.000214	0.950578	0.012877	0
84	1.051435	0.001670	0.948758	0.000264	0.948758	0.015916	0
85	1.057865	0.001670	0.946464	0.000339	0.946464	0.020492	0

TABLE 3.3 has been appended from 180 to 85 lines.